



Kauno technologijos universitetas

Informatikos fakultetas

Netolygiai pasiskirsčiusių duomenų klasifikavimo metodų tyrimas monetų vaizdams

Magistro baigiamojo laipsnio projektas

Mantas Šutas

Projekto autorius

Dr. Eglė Butkevičiūtė

Vadovė



Kauno technologijos universitetas

Informatikos fakultetas

Netolygiai pasiskirsčiusių duomenų klasifikavimo metodų tyrimas monetų vaizdams

Magistro baigiamojo laipsnio projektas

Dirbtinio intelekto informatika, 6211BX007

Informatikos mokslai, Informatika (B01)

Mantas Šutas

Projekto autorius

Dr. Eglė Butkevičiūtė

Vadovė

Prof. Andrius Kriščiūnas

Recenzentas



Kauno technologijos universitetas

Informatikos fakultetas

Mantas Šutas

Netolygiai pasiskirsčiusių duomenų klasifikavimo metodų tyrimas monetų vaizdams

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autorius 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.

Mantas Šutas

Patvirtinta elektroniniu būdu



Kauno technologijos universitetas

Informatikos fakultetas

Magistro studijų baigiamasis projektas

Projekto tema

Netolygiai pasiskirsčiusių duomenų klasifikavimo metodų tyrimas monetų
vaizdams

Reikalavimai ir sąlygos
(prireikus pavadinimas gali
būti patikslintas)

Vadovas

Dr. Eglė Butkevičiūtė

2026

(pareigos, vardas, pavardė, vadovo parašas)

(data)

Šutas, Mantas. Netolygiai pasiskirsčiusių duomenų klasifikavimo metodų tyrimas monetų vaizdams. Magistro baigiamasis projektas / vadovė Dr. Eglė Butkevičiūtė; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų programa: Dirbtinio intelekto informatika, 6211BX007.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos mokslai, Informatika (B01).

Reikšminiai žodžiai: monetų klasifikavimas, netolygūs duomenys, vaizdų paieška, požymių vektoriai, kandidatų perrikiavimas, patikimumas.

Kaunas, 2026. 128 p.

Santrauka

Tiriama, kaip nustatyti monetos vertę iš nuotraukų, kai rinkinyje vienu monetų pavyzdžių daug, kitų mažai, o skirtingos vertės monetos dažnai atrodo panašiai. Vietoje vieno tiesioginio klasifikatoriaus darbe taikoma dviejų etapų paieška. Iš monetos vaizdų pirmiausia sudaromas galimų kandidatų sąrašas, o tada jo tvarka tikslinama pagal svorį, skersmenį ir metų laikotarpį.

Naudotas rinkinys, kuriame prie monetų vaizdų pateikti papildomi atributai. Vaizdai paverčiami požymių vektoriais, pagal juos ieškoma artimiausių mokymo pavyzdžių, o papildomi atributai tik keičia kandidatų vietą sąraše. Jie nėra naudojami kaip taisyklė, kuri iš karto pašalina kandidatą. Rezultatai vertinti pagal makro F1, atkūrimą@K, dažnų, vidutinio dažnio ir retų klasių grupes bei patikimumo patikras.

Eksperimentuose vien vaizdo paieška nebuvo pakankama vienam galutiniam atsakymui. Vien kNN paieška pagal vaizdą pasiekė makro F1 0,4630. Kai tas pats kandidatų sąrašas buvo perrikiuotas pagal papildomus atributus, pagrindinis variantas pasiekė makro F1 0,8190 ir atkūrimą@5 0,9813. Metų laikotarpis turėjo didžiausią poveikį kandidatų tvarkai, o svoris ir skersmuo padėjo tada, kai jų reikšmės buvo žinomos.

Sudėtingesni mokomi perrikiuotojai šiame rinkinyje nebuvo stabilesni už paprastas paaiškinamas taisykles. Dėl to galutiniame sprendime paliktas taisyklinis perrikiavimas ir atskira patikimumo patikra. Ši patikra parodo, kada pirmų kandidatų balų skirtumas per mažas ir vieno atsakymo geriau nepateikti.

Šutas, Mantas. Study of Classification Methods for Imbalanced Data in Coin Images. Master's Final Degree Project / supervisor Dr. Eglė Butkevičiūtė; Faculty of Informatics, Kaunas University of Technology.

Study programme: Artificial Intelligence Informatics, 6211BX007.

Study field and area (study field group): Informatics Sciences, Informatics (B01).

Keywords: coin classification, imbalanced data, image retrieval, feature vectors, candidate reranking, reliability.

Kaunas, 2026. 128 pages.

Summary

The thesis examines coin denomination identification from images when some denominations have many examples, others have very few, and visually similar coins may belong to different classes. Instead of using one direct classifier, the solution uses two retrieval stages. Coin images first form a candidate list, and the candidate order is then adjusted using weight, diameter and year range information.

The data set contains coin images and additional attributes. Images are converted into feature vectors, nearest training examples are retrieved, and the additional attributes only change the candidate order. They are not used as a rule that immediately removes a candidate. The results are evaluated using macro F1, recall@K, frequent, medium-frequency and rare class groups, and reliability metrics.

In the experiments, image retrieval alone was not sufficient for one final answer. Image-based kNN retrieval reached macro F1 of 0.4630. When the same candidate list was reranked using additional attributes, the main variant reached macro F1 of 0.8190 and recall@5 of 0.9813. Year range information had the strongest effect on the candidate order, while weight and diameter helped when their values were available.

More complex learned rerankers were not more stable on this data set than simple explainable rules. Therefore, the final solution keeps rule-based reranking and a separate reliability check. This check shows when the score difference between the top candidates is too small and one final answer should not be returned.

Turinys

Lentelių sąrašas	10
Paveikslų sąrašas	12
Terminų sąrašas	14
Įvadas	15
Projekto naujumas ir aktualumas	15
Tikslas ir uždaviniai	16
Dokumento struktūra.....	16
1. Monetų vaizdų klasifikavimo ir netolygių duomenų literatūros analizė	18
1.1. Problema, netolygūs duomenys ir vertinimas	18
1.2. Požymių vektorių požiūris ir kandidatų paieška	23
1.3. Papildoma informacija ir kandidatų perrikiavimas	30
1.4. Patikimumas, kada sistema neturėtų pateikti atsakymo, ir aprėptis.....	35
1.5. Literatūros analizės išvados	37
2. Monetų kandidatų paieškos sistemos projektas	43
2.1. Statinis sistemos vaizdas	43
2.2. Dinaminis sistemos vaizdas	44
2.3. Komeracinė specifikacija.....	49
2.4. Sistemos funkcijos	49
3. Monetų atpažinimo naudotojo sąsaja	51
4. Monetų kandidatų paieškos sprendimo projektavimas ir įgyvendinimas	55
4.1. Kontekstas ir sprendimo idėja.....	55
4.2. Reikalavimų specifikacija	56
4.3. Bendroji sprendimo architektūra.....	57
4.4. Duomenų analizė	58
4.5. Duomenų paruošimas ir pateikimas sistemai.....	64
4.6. Požymių vektorių išgavimas	67

4.7. Kandidatų paieška pagal požymių vektorius	70
4.8. Papildomos informacijos perrikiavimas	72
4.9. Patikimumas ir kada sistema neturėtų pateikti atsakymo	75
4.10. Projektavimo ir įgyvendinimo išvados	76
5. Monetų atpažinimo eksperimento rezultatai.....	78
5.1. Tikslai ir tikrinamos prielaidos	78
5.2. Vertinimo tvarka	78
5.3. Eksperimentų palyginimo tvarka	79
5.4. Eksperimentai ir rezultatai	80
5.4.1. Bazinis klasifikatorius.....	84
5.4.2. Paieška pagal vaizdo panašumą be papildomos informacijos	85
5.4.3. Papildomos informacijos perrikiavimo patikros	88
5.4.4. Kandidatų rinkinio dydžio įtaka.....	91
5.4.5. Patikimumas ir kada sistema neturėtų pateikti atsakymo	94
5.4.6. Kelių pavyzdžių režimas ir laipsniškas klasių pridėjimas	96
5.4.7. Mokomas modelis kandidatų tvarkai keisti	97
5.4.8. Etikečių smulkumo patikra	99
5.4.9. Klaidų analizė	102
5.5. Eksperimentinės dalies išvados.....	107
5.6. Kas gali riboti rezultatų patikimumą.....	108
6. Išvados.....	109
7. Ribojimai ir tolesni darbai	110
8. Literatūros sąrašas.....	111
Priedai	115
1 priedas. Bandomųjų variantų ir skaičiavimų santraukos	115
1.1. Bazinis nustatymų rinkinys.....	115
1.2. Požymių vektorių mokymo pagal panašumą bandomieji variantai	115
1.3. Paprastos paieškos taisyklės.....	116

1.4. Kandidatų sąrašo įvairovės didinimas.....	116
1.5. Paieškos rinkinio retinimas	116
1.6. Dviejų pusių patikra	117
1.7. Mokomas modelis kandidatų tvarkai keisti su HGB	117
1.8. Kiti perrikiavimo ir statistinių korekcijų variantai.....	117
1.9. Atsparumo papildomos informacijos trūkumui patikra	118
1.10. Kelių pavyzdžių ir laipsniškas scenarijus	118
1.11. Patikimumo balo pavertimas į tikimybę	119
1.12. Dviejų pusių požymių vektorių sujungimo svoris α	119
1.13. Bandymų sugrupavimas pagal temas	119
Paprastos paieškos taisyklės.....	116
1.14. Svarbiausių bandymų grupių paaiškinimai	121
1.15. Visų bandymų sąrašas	123
1.16. Vien vaizdo informacijos pakankamumas	128
1.17. Kandidatų sąrašo formavimo strategijos.....	128
1.18. Vaizdo požymių gerinimo bandymai.....	129
1.19. Mokomas modelis kandidatų tvarkai keisti ir jo stabilumas.....	129
1.20. Patikimumo ir neatsakymo analizė	130
1.21. Mastelio ir stabilumo kompromisai	130

Lentelių sąrašas

1.1 lentelė. Problemos, netolygių duomenų ir vertinimo literatūros suvestinė	21
1.2 lentelė. Požymių vektorių požiūrio ir kandidatų paieškos literatūros suvestinė.....	29
1.3 lentelė. Papildomos informacijos ir kandidatų perrikiavimo literatūros suvestinė	34
1.4 lentelė. Patikimumo ir atvejų, kai sistema neturėtų pateikti atsakymo, literatūros suvestinė	37
1.5 lentelė. Literatūros analizės suvestinė	40
4.1 lentelė. Konteksto ir sprendimo idėjos suvestinė	56
4.2 lentelė. Duomenų analizės suvestinė	64
4.3 lentelė. Duomenų paruošimo ir jų formos suvestinė	67
4.4 lentelė. Projektavimo dalies suvestinė.....	77
5.1 lentelė. Pagrindinių eksperimentų rezultatų suvestinė	82
5.2 lentelė. Komponentų ir ribojimų patikrų suvestinė	83
5.3 lentelė. Patikimumo rezultatų suvestinė, kai daliai atvejų sistema atsakymo nepateikia.....	84
5.4 lentelė. Bazinio klasifikatoriaus rezultatų suvestinė.....	84
5.5 lentelė. Paieškos pagal vaizdo panašumą ir susijusių patikrų rezultatų suvestinė.....	85
5.6 lentelė. Požymių vektorių išgavimo modelių palyginimo suvestinė	87
5.7 lentelė. Papildomos informacijos perrikiavimo patikrų rezultatų suvestinė.....	88
5.8 lentelė. Kandidatų sąrašo dydžio, retinimo ir vaizdų apdorojimo patikrų suvestinė.....	92
5.9 lentelė. Patikimumo ir neatsakymo patikrų rezultatų suvestinė	94
5.10 lentelė. Kelių pavyzdžių režimo ir laipsniško klasių pridėjimo rezultatų suvestinė	96
5.11 lentelė. Mokomo perrikiavimo bandymų rezultatų suvestinė	97
5.12 lentelė. Etikečių smulkumo patikros rezultatų suvestinė.....	100
1.1 lentelė. Kelių pavyzdžių scenarijaus rezultatų suvestinė.....	118
1.2 lentelė. Bandymų sugrupavimo pagal temas suvestinė	119
1.3 lentelė. Visų bandymų sąrašas.....	123
1.4 lentelė. Vien vaizdo informacijos pakankamumo rezultatų suvestinė	128
1.5 lentelė. Kandidatų sąrašo formavimo strategijų rezultatų suvestinė	128
1.6 lentelė. Vaizdo požymių gerinimo bandymų rezultatų suvestinė.....	129

1.7 lentelė. Mokomo modelio kandidatų tvarkai keisti rezultatų suvestinė	129
1.8 lentelė. Patikimumo ir neatsakymo analizės rezultatų suvestinė.....	130
1.9 lentelė. Mastelio ir stabilumo kompromisų rezultatų suvestinė	130

Paveikslų sąrašas

2.1 pav. Monetų klasifikatoriaus sistemos diegimo diagrama.....	43
2.2 pav. Monetų klasifikatoriaus sistemos klasių diagrama	43
2.3 pav. Modelio atnaujinimo sekos diagrama	45
2.4 pav. Modelio mokymo sekos diagrama	46
2.5 pav. Vaizdo įkėlimo sekos diagrama	47
2.6 pav. Monetų klasifikavimo sekos diagrama	48
2.7 pav. Naudotojo atsiliepimo pateikimo sekos diagrama	49
2.8 pav. Monetų klasifikatoriaus sistemos atvejų diagrama	50
3.1 pav. Pagrindinis puslapis su naudotojo navigacijos pasirinkimais.....	51
3.2 pav. Monetos įkėlimo forma su papildomų duomenų pateikimu	52
3.3 pav. Monetos įkėlimo puslapis, po įkėlimo gauti rezultatai	52
3.4 pav. Rezultatų puslapis, kuriame pateikiama monetų klasifikacija.....	53
3.5 pav. Rezultatų puslapis, monetos informacija ir atsiliepimo palikimas	53
3.6 pav. Atsiliepimų peržiūros ir redagavimo puslapis	54
3.7 pav. Atsiliepimų redagavimas	54
4.1 pav. Sprendimo srauto schema	58
4.2 pav. Monetų identifikatorių pasikartojimų histograma duomenų rinkinyje.....	59
4.3 pav. Monetų svorio pasiskirstymo histograma	60
4.4 pav. Monetų skersmenų pasiskirstymo histograma	60
4.5 pav. Klaidų sumaišymo matrica	61
4.6 pav. Sklaidos diagrama, kurioje lyginama iš svorio, skersmens ir metų laukų apskaičiuota monetos vidutinė data ir tikroji vidutinė data	62
4.7 pav. Sklaidos diagrama, kurioje monetų skaitiniai laukai, taikant pagrindinių komponentų analizę, sumažinti iki dviejų ašių	62
4.8 pav. Sklaidos diagrama, kurioje tie patys monetų skaitiniai laukai t-SNE būdu išdėstyti dvimatėje erdvėje.....	63
4.9 pav. Monetų svorio ir skersmens dėžučių diagramos.....	63
5.1 pav. (1 dalis). Duomenys, vaizdo modeliai ir vaizdo pavertimas skaičiais palyginimui	81

5.2 pav. Galimų klaidų vietos sistemos eigoje	101
5.3 pav. Klaidų analizės pavyzdžiai I	104
5.4 pav. Klaidų analizės pavyzdžiai II.....	105

Terminų sąrašas

Terminas	Paaiškinimas
koduotuvas	Modelio dalis, kuri vaizdą paverčia požymių vektoriumi.
CNN	Konvoliucinis neuroninis tinklas (angl. convolutional neural network). Vaizdų modelio tipas, kuris iš vaizdo išgauna svarbius požymius.
RDF	Resource Description Framework. Duomenų aprašymo būdas, kai įrašai saugomi kaip tarpusavyje susijęs tinklas.
SPARQL	Užklausų kalba, skirta duomenims iš RDF tinklo ištraukti.
kNN	k-nearest neighbors. Artimiausių kaimynų paieška, kai ieškoma labiausiai į užklausą panašių pavyzdžių.
PCA	Pagrindinių komponentų analizė. Būdas sumažinti svorio, skersmens ir metų informaciją iki kelių ašių, kad būtų lengviau pamatyti bendrą jų struktūrą.
t-SNE	Būdas parodyti panašius įrašus dvimatėje erdvėje taip, kad panašūs įrašai atsidurtų arčiau vieni kitų.
vektorinis įterpinys	Skaitinis objekto atvaizdavimas vektoriumi, naudojamas panašumui skaičiuoti.
požymių vektorius	Skaičių seka, aprašanti objekto savybes ir naudojama objektams palyginti.
klasifikatorius	Modelis arba taisyklė, priskirianti objektą vienai iš klasių.
makro F1	Rodiklis, kai F1 apskaičiuojamas kiekvienai klasei atskirai, o tada apskaičiuojamas jų vidurkis.
atkūrimas@K	Rodiklis, parodantis, ar teisingas variantas pateko į pirmus K pasiūlymų.
kiek užklausų sistema atsako	Parodo, į kokią dalį užklausų sistema apskritai pateikia atsakymą.
ECE	Vidutinė tikimybių paklaida. Parodo, kiek modelio tikimybės atitinka realų tikslumą.
MCE	Didžiausia tikimybių paklaida.
AURC	Plotas po kreive, kuri vienu metu rodo klaidų riziką ir kiek užklausų sistema atsako. Parodo, kaip gerai patikimumo signalas atskiria lengvesnius ir sunkesnius atvejus.

Ivadas

Projekto naujumas ir aktualumas

Monetos vertę sunku nustatyti vien iš vaizdų, nes realūs monetų vaizdai būna nevienodos kokybės, o duomenų rinkinys nėra tolygus. Vienu verčių pavyzdžių daug, kitų labai mažai, o skirtingų verčių monetos kartais skiriasi tik smulkiomis detalėmis. Todėl neužtenka gauti vieną bendrą tikslumo skaičių. Reikia matyti, ar sistema padeda ir retoms klasėms, ar ji gerai veikia tik ten, kur mokymo pavyzdžių daug.

Tokia problema svarbi tada, kai sistema turi padėti žmogui rasti monetos vertę, o ne tik parinkti vieną klasę uždarame pavyzdžių rinkinyje. Jei kelios monetos atrodo panašiai, naudingiau parodyti kelis galimus kandidatus ir aiškiai pasakyti, kada pirmu atsakymu pasitikėti per rizikinga.

Literatūros analizėje matyti, kad artimi darbai dažnai nagrinėja tik dalį šios problemos. Netolygaus mokymo darbai dažniausiai vertina vieno atsakymo klasifikaciją. Monetų paieškos darbai dažnai pateikia kelis kandidatus, bet mažiau nagrinėja, kada sistema neturėtų pateikti vieno atsakymo. Kelių informacijos šaltinių sujungimo darbai dažnai aiškina sujungimo būdus, bet ne visada tikrina juos kartu su kandidatų rikiavimu.

Naujumas čia reiškia ne naują algoritmą, o tai, kad toje pačioje monetos vertės nustatymo sistemoje patikrinama vaizdo kandidatų paieška, papildomų atributų perrikiavimas ir atskira patikimumo patikra. Taip darbas ne tik pasirenka vieną sprendimo variantą, bet ir atskiria, ar klaida atsiranda pradinėje paieškoje, kandidatų tvarkos keitime ar sprendime pateikti vieną atsakymą.

Pirmiausia pagal priekinės ir galinės pusės vaizdus randami panašūs kandidatai. Tada jų tvarka keičiama pagal svorį, skersmenį ir metų intervalą. Galiausiai tikrinama, ar pirmas kandidatas pakankamai atsiskiria nuo kitų, kad sistema galėtų pateikti vieną atsakymą.

Atskiras vertinimas parodo, kur atsiranda klaida. Vienu atveju teisinga moneta visai nepatenka į kandidatų sąrašą. Kitu atveju ji randama, bet po perrikiavimo lieka ne pirmoje vietoje. Dar kitu atveju sistema pateikia atsakymą, nors keli kandidatai yra labai arti vienas kito.

Dėl to rezultatai rodomi ne vien bendru tikslumu. Naudojamas makro F1, dažnų, vidutinio dažnio ir retų klasių grupės, atkūrimas@5, rizikos ir atsakytų užklausų dalies patikra bei klaidų analizė. Toks vertinimas leidžia matyti, ar pagerėjimas neatsiranda tik iš dažnų klasių arba iš to, kad sistema daliai užklausų nepateikia atsakymo.

Papildomi monetos laukai naudojami atsargiai. Svoris, skersmuo ir metų intervalas gali padėti atskirti vizualiai panašius kandidatus, bet šie laukai gali būti nepilni arba netikslūs. Todėl kandidatai neatmetami vien dėl šių laukų neatitikimo. Jie tik koreguoja jau rastų kandidatų tvarką.

Be pagrindinio sprendimo, darbe patikrintos ir sudėtingesnės literatūroje minimos kryptys. Jos lyginamos su pagrindiniu variantu, kad būtų matyti, ar sudėtingesnis požymių mokymas, kaimynų sutapimo tikrinimas arba mokomas perrikiavimas iš tikrųjų pagerina rezultatą šiame rinkinyje. Kai šie variantai rezultato nepagerina, pagrindiniu sprendimu lieka paprastesnis perrikiavimas, kurio poveikį lengviau paaiškinti per klaidų analizę.

Tyrimas sutelktas į monetos vertės nustatymą iš vaizdų ir kelių papildomų laukų. Monetos teksto atpažinimas vaizde, naujos klasių hierarchijos kūrimas ir platesnis etikečių pertvarkymas paliekami kaip atskiros kryptys. Tokia riba leidžia tiksliai patikrinti, ar išskaidyta paieškos, perrikiavimo ir

patikimumo eiga pagerina vien vaizdu paremtą atsakymą ir parodo, kur atsiranda klaida. Teisinga moneta gali būti nerandama kandidatų sąrašė, gali likti ne pirmoje vietoje arba pirmas atsakymas gali būti per rizikingas.

Tikslas ir uždaviniai

Tyrimo objektas yra monetos vertės nustatymo sistema, kuri pirmiausia pagal vaizdus suranda kelis panašiausius variantus, o tada jų tvarką sąrašė patikslina pagal papildomus atributus (svorį, skersmenį ir metų laikotarpį).

Darbo tikslas yra iširti ir įvertinti, kaip netolygiame monetų duomenų rinkinyje monetos vertės nustatymą veikia dviejų etapų sprendimas, kuriame pagal vaizdą sudaromas galimų kandidatų sąrašas, o jo tvarka tikslinama pagal svorio, skersmens ir metų laikotarpio duomenis.

1. Įvertinti, ar vaizdo paieška gali būti patikimas pirmas etapas ir ar teisinga moneta pakankamai dažnai patenka į kandidatų sąrašą, kad vėlesnis perrikiavimas dar galėtų ją iškelti į pirmą vietą.
2. Patikrinti, kaip vaizdo dalies pasirinkimai veikia pradinę paiešką, ar reikia abiejų monetos pusių, kaip keičiasi rezultatas su skirtingais vaizdo koduotojais ir ar vien vaizdo informacijos pakanka galutiniam atsakymui.
3. Įvertinti, kaip svoris, skersmuo ir metų laikotarpis keičia kandidatų tvarką, kurie laukai duoda didžiausią naudą ir kada papildoma informacija tampa rizikinga dėl trūkstanų arba netikslių reikšmių.
4. Palyginti pagrindinį taisyklėmis paremtą perrikiavimą su sudėtingesnėmis alternatyvomis, tokiomis kaip mokomi perrikiuotojai, kiti požymių mokymo tikslai, kandidatų sąrašo dydžio keitimas ir kandidatų įvairovės didinimas.
5. Patikrinti, ar sprendimą galima naudoti atsargiau, nustatant, kada sistema turėtų nepateikti vieno galutinio atsakymo ir kaip ji veikia, kai naujoms ar retoms klasėms pridedami tik keli pavyzdžiai.
6. Išanalizuoti likusias klaidas pagal sprendimo grandinės vietą ir nustatyti, ar teisinga klasė nerandama paieškoje, ar lieka ne pirmoje vietoje po perrikiavimo, ar klaidą lemia trūkstami laukai, artimos klasės arba per smulkios etiketės.

Dokumento struktūra

Dokumentas sudarytas iš įvado, literatūros analizės, sistemos ir sprendimo projektavimo, eksperimentų, išvadų ir priedų.

Literatūros analizėje aptariami netolygių duomenų vertinimo, požymių vektorių, kandidatų paieškos, papildomos informacijos perrikiavimo ir patikimumo darbai.

Sistemos projektavimo ir naudotojo sąsajos skyriai aprašo, kaip sistema priima užklausą, saugo duomenis, pateikia kandidatų sąrašą ir leidžia naudotojui peržiūrėti rezultatą.

Sprendimo projektavimo ir įgyvendinimo skyriuje aprašoma visa atpažinimo eiga nuo duomenų paruošimo iki kandidatų paieškos, perrikiavimo ir atsakymo nepateikimo taisyklės.

Eksperimentų skyriuje tikrinama, kiek padeda vaizdo paieška, papildomi monetos laukai, kandidatų sąrašo dydis, patikimumo riba ir kiti palyginti variantai.

Išvadose apibendrinama, ką parodė eksperimentai, kokie yra ribojimai ir kokias patikras verta atlikti toliau.

Prieduose pateikiamos bandomųjų variantų ir papildomų skaičiavimų santraukos, kad pagrindinis tekstas nebūtų perkrautas.

1. Monetų vaizdų klasifikavimo ir netolygių duomenų literatūros analizė

1.1. Problema, netolygūs duomenys ir vertinimas

Realiose taikymo srityse duomenų pasiskirstymas pagal klases dažnai yra stipriai netolygus. Nedidelė dalis klasių turi daug pavyzdžių, o didelė dalis klasių turi labai mažai. Literatūroje toks pasiskirstymas apibūdinamas kaip netolygus (angl. long-tailed). Tai reiškia, kad kelios klasės turi daug pavyzdžių, o dauguma klasių turi labai mažai. Dėl to bendri vidurkiai, pavyzdžiui, bendras tikslumas, gali paslėpti, kaip sistema veikia retose klasėse, nors būtent ten ji dažnai ir klysta. Toks netolygumas dažnai nėra vien duomenų klaida. Darbe apie objektų atpažinimą netolygiame vaizdų rinkinyje pabrėžiama, kad taip dažnai nutinka ir realiuose rinkiniuose. Kitaip tariant, didelis netolygumas dažnai yra ne atsitiktinumas, o įprasta tokių duomenų savybė [1].

Kai dauguma klasių turi labai mažai pavyzdžių, svarbu suprasti, kodėl atsiranda retų klasių klaidos. Viena iš aiškiai suformuluotų priežasčių yra požymių iškraipymas. Darbe apie mokymą be polinkio į dažnas klases (angl. unbiased shared feature learning) rašoma, kad kai duomenys labai netolygūs, modelis geriau išmoka dažnų klasių požymius. Dėl to retos klasės gauna silpnesnius požymius, todėl jų balai dažniau lieka per maži. Dėl to net ir tada, kai užklausa priklauso retai klasei, dažnos klasės balas gali būti didesnis už teisingos retos klasės balą. Ši įžvalga šiam darbui svarbi, nes ji paaiškina, kodėl vien tik daugiau duomenų arba didesnis modelis ne visada išsprendžia problemą ir kodėl rezultatus retose klasėse reikia vertinti atskirai [2].

Vertinant netolygų mokymą, literatūra pabrėžia, kad vienos metrikos dažniausiai nepakanka. Reikia aiškiai parašyti, ką tiksliai matuoja pasirinktos metrikos, ir atskirai parodyti rezultatus grupėse pagal klasių dažnumą. Tam klasės skirstomos į dažnas, vidutines ir retas (angl. head, mid, tail) grupes. Šiame darbe grupės apibrėžiamos pagal tai, kiek mokymo pavyzdžių turi klasė. Dažnos klasės turi bent 100 pavyzdžių, vidutinės turi 20-99 pavyzdžius, o retos turi 1-19 pavyzdžių. Tai reikalinga tam, kad matytume, ar metodas tikrai pagerina rezultatus retose klasėse, ar tik pakelia bendrą vidurkį dažnų klasių sąskaita.

Šį argumentą sustiprina ir konkrečių rinkinių pavyzdžiai. Straipsnyje apie netolygų vaizdų atpažinimą minima, kad tipiniuose netolygiuose rinkiniuose kai kurios klasės turi vos kelis pavyzdžius, pavyzdžiui, ImageNet-LT turi klasių su 5 pavyzdžiais. Tai svarbu kaip aiškus ribojimas. Jei retose klasėse turime tik 1–10 pavyzdžių, sprendimo stabilumas retose klasėse tampa vienu svarbiausių klausimų, o vien bendras mokymo nuostolio vidurkis tokios problemos neišsprendžia [3].

Netolygių duomenų metodus literatūroje patogu skirstyti pagal tai, ką jie keičia. Vieni keičia duomenis. Kiti keičia nuostolį ar svorius. Dar kiti keičia pačių požymių vektorių erdvę arba klasifikatorių. Toks skaidymas šiame darbe naudingas todėl, kad leidžia aiškiai parodyti, kur tikimasi pagerėjimo. Kartu svarbu matyti ir konkretų pavyzdį. Straipsnyje apie požymių perkėlimą iš dažnų klasių į retas siūloma dalį dažnų klasių požymių perkelti retoms klasėms. Toks metodas labiau tinka klasifikatoriaus korekcijai, kai sprendimo ribos aiškios. Šiame darbe sprendimas formuluojamas kaip kelių geriausių kandidatų paieška, todėl ši kryptis čia svarbi tik kaip palyginimo taškas, o ne kaip tiesioginis pagrindinio sprendimo pakaitalas [4].

Kitas pavyzdys yra hierarchinis konvoliucinis neuroninis tinklas (angl. convolutional neural network, CNN) su žinių papildymu (angl. knowledge complementation). Straipsnyje aprašomas hierarchinis tinklas su bendrais ir užduočiai skirtais sluoksniais. Jis mokomas trimis lygiais. Pirmas lygis yra pati klasė. Antras lygis yra platesnė klasių grupė, kuriai priklauso kelios konkrečios klasės. Trečias lygis yra iš panašių duomenų sudaryta klasių grupė, kurioje kartu mokomos panašiai atrodančios klasės. Taip daugiau naudingos informacijos perkeliama į retas klases [5]. Tai tiesiogiai rodo, kad rezultatai

retose klasėse dažnai pagerėja įvedant papildomą struktūrą, tai yra hierarchiją, ir papildomas etiketes ar ryšius. Monetų atveju hierarchija, pavyzdžiui, vertė, valdovas ir monetos tipas, teoriškai egzistuoja, tačiau praktikoje ji gali būti nepilna arba nevienodai anotuota, todėl šios krypties perkėlimui reikėtų daugiau paruoštų duomenų ir papildomų ryšių tarp klasių.

Palyginimui verta aptarti ir dviejų etapų mokymą, kai pirmiausia mokomas vaizdo aprašas, o po to atskirai pertreniruojamas vieno atsakymo klasifikatorius. LORT straipsnyje siūlomas toks pertreniravimas. Įprastame mokyme modeliui pateikiamas paprastas atsakymas. Teisinga klasė turi gauti 1, o visos kitos klasės turi gauti 0. LORT keičia šias mokymo metu pateikiamas reikšmes. Teisingai klasei vis tiek paliekama didžiausia dalis, bet maža dalis paskirstoma ir kitoms klasėms, kad retos klasės mokymo metu nebūtų taip lengvai nustumiamos į šalį. Tokia kryptis svarbi, nes pagerėjimas ne visada atsiranda iš dar vieno vaizdo modelio. Kartais jis atsiranda iš pakeistos sprendimo taisyklės. Šio darbo sprendimas formuluojamas kaip kelių kandidatų paieška, todėl LORT labiau tinka kaip papildoma klasifikatoriaus alternatyva, o ne kaip pagrindinis kelias.

Straipsnyje taip pat pabrėžiama, kad tokiam mokymui nereikia iš anksto tiksliai žinoti klasių dažnių. Monetų atveju tai būtų prasminga, jei dalį uždavinio spręstume kaip klasifikaciją, pavyzdžiui, iki vertės ar valdovo lygio. Tada toks pertreniravimas galėtų mažinti retų klasių klaidas. Vis dėlto tam reikėtų atskiro klasifikatoriaus etapo ir aiškiai sutarto detalumo lygio [6].

Svarbus ir ryšys tarp netolygios problemos ir požymių vektorių mokymo. PaCo straipsnyje parodyta, kad net mokymas, kuris traukia panašius atvejus ir stumia skirtingus, gali labiau padėti dažnoms klasėms. Todėl siūlomi mokomi klasių vidurio taškai, kurie mokymą daro tolygesnį. Šiam darbui tai svarbu todėl, kad paieškoje teisingi kandidatai randami tik tada, kai panašios monetos pagal požymių vektorius yra arti viena kitos, o skirtingos monetos yra toliau. Ši įžvalga padeda suprasti, kada verta svarstyti PaCo tipo nuostolį. Tai verta daryti tada, kai retos klasės sistemingai pralaimi jau paieškos etape, o ne tik tada, kai trūksta duomenų [7].

Pažangesni netolygaus mokymo metodai šiame skyriuje veikia kaip galimos alternatyvos, o ne kaip automatiniai pasirinkimai. H2T (angl. head-to-tail feature fusion) reiškia požymių sujungimą iš dažnų klasių į retas klases, kurios turi mažai pavyzdžių. Toks metodas keičia klasifikatoriaus mokymą, kad retos klasės gautų daugiau požymių įvairovės. Dėl to jis būtų prasmingiausias tada, jei sprendime būtų atskiras klasifikatorius, kuris pateikia vieną klasę konkrečiame detalumo lygyje [4]. Hierarchinis žinių papildymas yra prasmingas tada, jei anotacijose yra patikima ir nuosekli hierarchija, tačiau jis aiškiai didina reikalavimus duomenims [5]. LORT tiesiogiai veikia klasifikatoriaus elgseną, o ne rikiavimo paiešką [6]. PaCo labiau artimas šiame darbe taikomai K geriausių kandidatų paieškai, nes jis skirtas mažinti polinkį dažnoms klasėms ir gali būti svarstomas tada, kai retos klasės išlieka silpnos jau paieškos etape [7]. Šiame darbe H2T, hierarchinių žinių papildymo ir LORT kryptys nebuvo įgyvendintos, nes jos reikalautų atskiro klasifikatoriaus etapo arba papildomų hierarchinių anotacijų ir ryšių, o pagrindinis sprendimas čia projektuojamas kaip K geriausių kandidatų paieška su papildomos informacijos perrikiavimu ir patikimumu.

Netolygių duomenų literatūra šiame darbe buvo reikalinga ne bendrai apžvalgai, o dviem aiškiems sprendimams pagrįsti. Pirma, ji parodė, kad rezultatų negalima vertinti vien bendru vidurkiu. Dėl to šiame darbe naudotos grupės pagal dažnumą ir rikiavimo metrikos. Antra, ji parodė, kad retų klasių problema savaime neišsprendžia vien padidinus modelį ar duomenų kiekį. Dėl to reikėjo atskiros retų klasių analizės.

Net ir tinkamos metrikos gali būti suprastos klaidingai, jei nepaaiškinama, ką jos iš tikrųjų matuoja. Praktikoje vienos metrikos neužtenka. Pavyzdžiui, makro F1 leidžia pamatyti vaizdą per klases, bet

tampa jautri, kai klasė turi labai mažai pavyzdžių. Atkūrimas@K atsako į kitą klausimą. Jis parodo, ar teisinga klasė bent patenka tarp pirmųjų siūlomų variantų. Tačiau vien jis nepasako, ar klaidos labiausiai kaupiasi retose klasėse, ar dažnose. Dėl to šiame darbe atkūrimas@K visada aiškinamas kartu su grupėmis pagal dažnumą.

Dar vienas svarbus akcentas yra skirtumas tarp uždaro rinkinio klasifikacijos ir kelių geriausių kandidatų scenarijaus. Netolygių duomenų literatūra dažnai nagrinėja vieno atsakymo klasifikaciją, todėl daugiausia dėmesio skiria nuostolių svoriams ir retų klasių balansavimui. Šiame darbe situacija kitokia. Čia svarbu ne tik vienas atsakymas, bet ir tai, ar teisinga klasė apskritai patenka į kandidatų sąrašą. Dėl to literatūros įžvalgas reikia pritaikyti atsargiai, o ne perkelti tiesiogiai.

Galiausiai, netolygūs uždaviniai dažnai susiję ir su tuo, kaip suskaidytos etiketės. Jei klasių ribos per smulkios, dalis klaidų gali atsirasti ne dėl modelio silpnumo, o dėl per smulkaus žymėjimo. Dėl to ši analizė aiškiai skiria du klausimus, kurie vėliau tampa eksperimentų logika. Pirmas, ar teisingas kandidatas apskritai patenka į kandidatų rinkinį. Antras, ar patekęs į rinkinį jis pakyla aukščiau, kai pridedame papildomą informaciją ir patikimumo logiką.

Svarbi papildoma detalė netolygiame paieškos vertinime yra tai, kad net tinkamas rikiavimo rodiklis gali būti pateiktas taip, kad netyčia labiau iškeltų dažnas klases. Pavyzdžiui, atkūrimas@K yra rodiklis, skaičiuojamas kiekvienai užklausiai, todėl rinkinyje, kuriame dauguma klasių turi labai mažai pavyzdžių, jis labiau atspindi tas klases, kurios turi daugiau testavimo užklausių. Dėl to, be dažnų, vidutinio dažnio ir retų klasių grupių, verta aiškiai parodyti ir per-klasinių makro vidurkinimą rikiavimo metrikoms, kai pirmiausia rodiklis apskaičiuojamas kiekvienai klasei atskirai, o tik tada vidurkinamas per klases. Toks vidurkinimas leidžia mažinti užklausių kiekio efektą ir tiksliau parodyti, ar sistema iš tikrųjų padeda retoms klasėms, kurios praktiniame scenarijuje dažnai yra svarbiausios.

K geriausių kandidatų scenarijuje vien atkūrimo@K ne visada pakanka. Šis rodiklis parodo, ar teisingas kandidatas patenka tarp pirmųjų K pozicijų, tačiau nepasako, ar klaida buvo maža, ar kandidatas liko gerokai žemiau. Todėl literatūroje dar naudojamas vidutinis tikslumas (angl. Average Precision, AP) ir jo vidurkinimas per užklausas (mAP), nes šie rodikliai vertina visą kandidatų eilę. Šiame darbe AP tipo rodikliai nėra pagrindinė metrika, bet jie naudingi kaip papildoma patikra, ar metodas gerina ne tik pasirinktą K, bet ir bendrą rikiavimo kokybę [8].

Galiausiai, patikimumo vertinimui svarbūs ir duomenų skaidymo principai. Paieškos sistemoje ypač pavojinga, kai per labai artimus duplikatus ar tą patį objektą informacija nuteka į skirtingas aibes, nes tada paieška gali atrodyti dirbtinai lengva. Monetų kontekste tai reiškia, kad tas pats monetos egzempliorius neturi patekti ir į mokymą, ir į testavimą, pavyzdžiui, per to paties objekto priekinės ir galinės pusės vaizdus ar kelias to paties objekto fotografijas. Tą patį pabrėžia ir netolygaus vertinimo literatūra. Vertinimo tvarka turi būti aiški ir atkartojama. Testavimo aibė turi tikrinti, ar sistema geba atpažinti naujus, anksčiau nematytus pavyzdžius, o ne tik prisiminti jau matytus.

Taip pat svarbu aiškiai apibrėžti dažnų, vidutinio dažnio ir retų klasių grupes. Literatūroje jos dažniausiai skiriamos pagal mokymo pavyzdžių skaičių, nes būtent jis parodo, kiek mokymo pavyzdžių turi klasė. Jei keistume skaidymą, keistųsi ir grupių ribos. Tada eksperimentų palyginimas taptų miglotas. Monetų rinkinyje tai ypač svarbu, nes labai mažoms klasėms net vienas papildomas ar prarastas pavyzdys gali perkelti klasę į kitą grupę ir iškreipti išvadas.

Retų klasių rodiklius reikia aiškinti atsargiai. Kai klasė turi labai mažai testavimo pavyzdžių, net kelios klaidos gali stipriai pakeisti pagal tą klasę skaičiuojamą rezultatą. Todėl vien galutinis skaičius be konteksto gali sudaryti per gerą arba per blogą vaizdą. Netolygaus mokymo apžvalgos pabrėžia,

kad tokiu atveju reikia rodyti ne tik skaičius, bet ir klaidų vietas. Šiame darbe klaidos skiriamos pagal tai, ar teisingas kandidatas nepatenka į kandidatų rinkinį, ar patenka, bet lieka žemai, ar sistema pateikia klaidingą atsakymą su dideliu patikimumo signalu.

Paieškoje svarbu ne tik užklausa, bet ir tai, iš kokių pavyzdžių ieškoma kandidatų. Jei kandidatai ieškomi tik tarp mokymo pavyzdžių, vertinimas labiau parodo, kaip gerai užklausa suranda tinkamus mokymo pavyzdžius. Kitas variantas būtų atskira palyginimo bazė, kuri nenaudojama mokant modelį, bet naudojama kandidatams rasti. Dėl to paieškos vertinimo tvarkoje būtina aiškiai pasakyti, kuris variantas naudojamas ir kaip išvengiama per lengvų sutapimų ar duplikatų.

Rezultatai šiame darbe rodomi dviem pjūviais. Vienas pjūvis aprašo vieno atsakymo kokybę per tikslumą, F1 pagal klasių dydį ir makro F1. Kitas pjūvis rodo kandidatų sąrašo kokybę per atkūrimą@K. Didelis netolygumas papildomai analizuojamas per dažnų, vidutinio dažnio ir retų klasių grupes, atskiriant paieškos, rikiavimo ir patikimumo klaidas.

1.1 lentelėje apibendrinta, kodėl šiam netolygiam uždaviniui neužtenka vien bendro rodiklio. Dėl to šiame darbe rezultatai skaidomi pagal dažnas, vidutinio dažnio ir retas klases, o klaidos atskiriamos pagal tai, ar jos atsirado paieškoje, ar vėlesniame rikiavime.

1.1 lentelė. Problemos, netolygių duomenų ir vertinimo literatūros suvestinė

Literatūra	Darbo santrauka	Reikšmė šiam darbui
Apžvalga ir straipsnis apie netolygų vaizdų rinkinį [2, 4]	Apžvalgoje ir straipsnyje parodyta, kad realiuose rinkiniuose vienu klasių yra daug, o kitų labai mažai. Taip pat aiškinama, kad bendras vidurkis dažnai paslepia, kaip sistema veikia retose klasėse.	Dėl to šiame darbe netolygumas laikomas pagrindine problema, o ne tik duomenų ypatybe.
Straipsnis apie bendrųjų požymių polinkį [2]	Straipsnyje aiškinama, kad dažnos klasės sukaupia daugiau naudingų požymių, todėl retos klasės pradeda atsilikti jau pačiame mokyme. Tai padeda suprasti, kodėl vien didesnis modelis ar daugiau duomenų nebūtinai išsprendžia problemą.	Dėl to retų klasių kokybė šiame darbe vertinama atskirai.
Straipsniai apie vertinimą ir testavimo tvarką [9, 10]	Šiuose straipsniuose pabrėžiama, kad vieno bendro rodiklio nepakanka. Reikia rodyti rezultatus pagal dažnas, vidutinio dažnio ir retas klases ir aiškiai pasakyti, kaip buvo atliktas vertinimas.	Dėl to šiame darbe naudotos dažnumo grupės, atskira retų klasių analizė ir aiški testavimo tvarka.
Straipsnis apie labai mažas retas klases [9]	Straipsnyje parodyta, kad netolygiuose rinkiniuose kai kurios klasės gali turėti vos kelis pavyzdžius. Tai svarbu todėl, kad tokiose klasėse rezultatai lengvai svyruoja, o bendras vidurkis gali atrodyti per geras.	Dėl to šiame darbe retos klasės vertinamos kaip viena svarbiausių rizikos vietų.
Straipsnis apie požymių perkėlimą iš dažnų klasių į retas [4]	Straipsnyje siūloma dalį informacijos, kuri gerai išmokstama iš dažnų klasių, panaudoti retoms klasėms. Toks kelias labiausiai tinka tada, kai sistema mokoma kaip vieno atsakymo klasifikatorius.	Ši kryptis nebuvo pasirinkta pagrindiniam sprendimui, nes bazinis vieno atsakymo klasifikatorius retoms klasėms veikė silpnai, o tolesnis darbas buvo perkeltas į kandidatų paiešką. Ji liko

		kaip palyginimo taškas, parodantis, kad retų klasių problemą reikia vertinti atskirai.
Straipsnis apie hierarchinį CNN su papildomomis žiniomis [5]	Straipsnyje retos klasės stiprinamos įvedant papildomas klasių grupes ir mokant modelį keliais lygiais. Taip dalis naudingos informacijos iš platesnių grupių perkeliama į smulkesnes klases.	Šis darbas parodė, kad klasių ryšiai gali padėti retoms klasėms, jei tie ryšiai yra aiškiai sužymėti. Šiame darbe tokia hierarchija nebuvo naudota kaip modelio dalis, nes duomenyse nėra pakankamai patikimo ir vienodo ryšio tarp vertės, valdovo, leidybos vietos ir konkrečios klasės. Dėl to ši idėja panaudota kaip argumentas vėliau tikrinti etikečių smulkumą, o ne kaip įgyvendintas modelis.
LORT straipsnis [6]	Straipsnyje vietoj griežto 1 ir 0 žymėjimo naudojamas švelnesnis tikimybių paskirstymas. Taip retos klasės mokymo metu nėra iš karto nustumiamos į šalį net tada, kai klasių dažniai nėra tiksliai žinomi.	Ši kryptis nebuvo pasirinkta pagrindiniam sprendimui, nes ji reikalautų iš naujo mokyti vieno atsakymo klasifikatorių ir keisti, kokias reikšmes mokymo metu gauna teisinga klasė ir kitos klasės. Tai būtų atskiras klasifikatoriaus tyrimas. Artimiausi praktiniai bandymai šiame darbe buvo ne LORT, o kiti vaizdo aprašų mokymo tikslai, tokie kaip FastAP, ProxyNCA ir PaCo tipo variantas. Šie bandymai pagrindinio rezultato nepagerino, todėl pagrindinis sprendimas buvo perkeltas į kandidatų paiešką ir perrikiavimą.
PaCo straipsnis [7]	Straipsnyje parodyta, kad požymių vektorių mokymas gali labiau padėti dažnoms klasėms, todėl siūlomi mokomi klasių centrai. Taip siekiama, kad dažnos klasės neturėtų tokio didelio pranašumo.	Ši idėja šiame darbe buvo patikrinta PaCo tipo mokymu. Testavimo dalyje jis pasiekė tik 0,7656 makro F1, kai pagrindinis kNN variantas su perrikiavimu pasiekė 0,8190. Todėl pilnas PaCo kelias nebuvo plėtotas toliau.
FastAP straipsnis [8]	Straipsnyje vidutinis tikslumas naudojamas ne tik galutiniam vertinimui, bet ir pačiam mokymui. Tai svarbu todėl, kad AP ir mAP geriau parodo visos kandidatų eilės kokybę negu vien atkūrimas ties vienu pasirinktu K.	Dėl to šiame darbe atkūrimas@K aiškinamas kartu su kitais kandidatų eilės rodikliais.

Literatūros analizė parodė, kad svarbu vertinti ne tik bendrą rezultatą, bet ir sistemos darbą retose klasėse. Todėl darbe naudojamos dažnumo grupės, aiški testavimo tvarka ir papildomi kandidatų eilės rodikliai. Taip pat matyti, kad retų klasių problemą galima spręsti keliais būdais, tačiau ne visi jie vienodai tinka kandidatų paieškai.

1.2. Požymių vektorių požiūris ir kandidatų paieška

Monetų identifikavimo uždavinys išsiskiria tuo, kad realiame kontekste klasių, ypač monetų tipų (angl. issue), skaičius gali būti labai didelis, o daugeliui klasių trūksta pakankamai gerų pavyzdžių. Romos respublikos monetų paieškos straipsnyje pabrėžiama, kad klasifikacija, kai sistema renkasi tik iš anksto apibrėžto klasių rinkinio (angl. closed-set classification), tampa nepraktiška, kai monetų tipų skaičius gali viršyti dešimtis tūkstančių ir daugumai jų nėra realu surinkti pakankamai pavyzdžių. Dėl to čia labiau tinka paieškos scenarijus. Sistema turi pasiūlyti kelis panašiausius kandidatus, o galutinis sprendimas patvirtinamas naudotojo ar kito tikrinimo etapo [11].

Straipsnis apie Romos respublikos monetų paiešką pateikia ir konkrečius masto skaičius, kurie leidžia tiksliau apibrėžti problemos ribas. Teigiama, kad vien Romos imperinių monetų tipų yra virš 50 000, o skaičius dar labiau auga įtraukiant kitas monetų grupes. Taip pat pateikiamas didelis duomenų rinkinys – 14 820 vaizdų ir 7 605 monetų tipų. Mokymui parenkama tik maža, gerai sužiūrėta poaibė. Treniruotei naudojami tik tie monetų tipai, kurie turi daugiau nei 20 pavyzdžių, ir taip sudaromas 542 vaizdų iš 24 monetų tipų mokymo rinkinys, o vertinimui minima 7 605 monetų tipų apimtis. Tai rodo, kad porinio palyginimo arba paieškos sprendimas gali būti mokomas ant mažesnės poaibės, bet taikomas labai didelei ir netolygiai kandidatų erdvei. Kartu tai yra ir ribojimas, nes mokymo poaibė čia nėra pilnai netolygi, todėl atskirai reikia įvertinti, kaip sistema veikia klasėse, kurioms turime labai mažai pavyzdžių.

Tokia logika tiesiogiai veda prie požymių vektorių požiūrio. Praktikoje modelis kiekvieną monetos vaizdą paverčia skaičių rinkiniu. Sistema lygina šiuos rinkinius tarpusavyje ir randa artimiausius pavyzdžius. Pagal juos sudaromas kelių galimų klasių kandidatų sąrašas, o ne iš karto pateikiamas vienas galutinis atsakymas. Praktikoje tai svarbu ir sistemos plėtrai, nes pavyzdžių rinkinį galima papildyti naujais pavyzdžiais ar klasėmis nepertreniruojant viso modelio nuo nulio. Kad toks požiūris nebūtų tik bendras aprašymas, jis vertinamas pagal aiškius kriterijus. Pirma, ar teisinga klasė apskritai patenka į kandidatų rinkinį. Antra, ar kandidatų tvarka yra pakankamai gera, ypač retoms klasėms. Trečia, kaip kokybę ir kainą veikia siūlomų kandidatų skaičius ir kandidatų rinkinio dydis.

Monetų atpažinimo kontekste papildomas, bet praktiškai labai svarbus aspektas yra tai, kad viena moneta paprastai aprašoma dviem vaizdais, monetos priekiu ir galu. Monetų domeno darbuose ši prielaida yra įprasta, nes informacija pusėse dažnai yra komplementari. Vienoje pusėje gali būti valdovo atvaizdas ar stambus simbolis, o kitoje gali būti monetos teksto fragmentų, vertės nuorodų ar kitų smulkių identifikacinių elementų. Todėl požymių vektorių požiūryje natūraliai kyla klausimas, kaip šią dviejų pusių informaciją integruoti taip, kad būtų išlaikyta ir diskriminacija, ir stabilumas netolygiame režime.

Šiame darbe dviejų pusių naudojimas yra svarbus ne kaip papildomas patogumas, o kaip metodinis pasirinkimas, susijęs su tiksliniu detalumo lygiu. Jei tikslas būtų stambesnė klasė, pavyzdžiui, denominacija, vienos pusės signalas dažnai būtų pakankamas. Tačiau smulkaus detalumo identifikavime, kur artimos klasės skiriasi smulkiomis detalėmis, vienos pusės informacija dažniau yra nepakankama. Dėl to abiejų pusių požymių vektorių panašumo balai šiame darbe sujungiami sprendimo lygyje, taikant vėlyvo sujungimo principą. Taip mažinama rizika, kad vienos pusės prastesnė kokybė ar uždengtos detalės nulems klaidingą kandidatų rikiavimą [12].

Svarbu pabrėžti, kad monetų atpažinimo literatūroje yra ir uždaro rinkinio klasifikacijos sprendimų, kurie pasiekia aukštą tikslumą [26]. Tačiau dažniausiai jie sprendžia mažesnio masto uždavinį. Pavyzdžiui, modernių Brazilijos monetų darbe aprašomas kelių etapų CNN sprendimas, kuris pasiekia apie 98 procentų tikslumą. Toks rezultatas svarbus kaip atskaitos taškas. Kai klasių mažiau

ir jos apibrėžtos stambiau, pavyzdžiui, pagal vertę, uždaro rinkinio požiūris gali būti pakankamas. Tačiau senovinių monetų scenarijuje klasių yra daug daugiau, jų žymėjimas smulkesnis, o pasiskirstymas netolygus. Dėl to šiame darbe pasirinkta paieškos formuluotė yra praktiškesnė [12].

Monetų paieškos sistemos logiką gerai parodo darbas su naudotojo sąsaja. Jame aiškiai aptariamas kompromisas tarp skaičiavimo kainos ir naudotojo patirties. Rodomas pirmasis kandidatas arba trumpas kandidatų sąrašas, o ne visi kandidatai. Tas pats monetų paieškos darbas svarbus ir tuo, kad leidžia K pasirinkimą pagrįsti ne vien metrika, bet ir realiu naudojimu. Kitaip tariant, svarbu ne tik kas teoriškai geriausia, bet ir kiek kandidatų žmogus iš tikrųjų gali peržiūrėti [11].

Naudotojo sąsajos darbo eksperimentai suteikia praktinį argumentą, kaip K, tai yra kiek variantų rodoma naudotojui, sieti su skaičiavimo kaina. Duomenų bazėje yra 60 klasių, o kiekviena klasė turi po 10 pavyzdinių (angl. exemplar) vaizdų, todėl blogiausiu atveju vienai užklausai tektų 600 vaizdo atitikimų. Dėl to autoriai praktiškai tiria, kiek palyginimų vienai klasei, tai yra M, pakanka ir kiek klasių rodyti, tai yra N. Monetų paieškos sąsajos straipsnyje nurodoma, kad didžiausias tikslumas (99 %) pasiekiamas, kai $M \geq 5$ ir $N \geq 7$, o praktinė rekomendacija yra $M = 5$ ir $N = 10$ dėl laiko ir naudos kompromiso. K geriausių kandidatų nėra vien teorinė metrika, o sistemos nustatymas, kurį reikia parinkti kartu su paieškos kaina ir naudotojo galimybe peržiūrėti rezultatus.

Kad K geriausių kandidatų vertinimas būtų korektiškas, reikia aiškiai apibrėžti (a) ką laikome teisingu atitikmeniu (motyvo lygis, vertės lygis, monetos leidybos tipo lygis), (b) kokį K laikome realistišku, ir (c) kokias metrikas naudojame. Netolygaus vertinimo literatūra pabrėžia, kad kai sprendimas yra kandidatų sąrašas, rikiavimo metrikos yra natūralus pasirinkimas, o rezultatų grupės pagal dažnį leidžia pamatyti uodegos elgseną. Praktiniam K geriausių kandidatų scenarijui dažniausiai minimas atkūrimas@K (angl. Recall@K), kuris tiesiogiai atsako į klausimą, ar teisingas kandidatas patenka į K pateikiamų kandidatų sąrašą.

Toliau kyla svarbus klausimas, kaip mokyti požymių vektorius taip, kad mokymo tikslas atitektų vertinimą. Literatūroje apie požymių vektorių mokymą pagal panašumą, dar vadinamą metric learning arba deep metric learning, dažniausiai mokoma taip, kad tos pačios klasės pavyzdžiai būtų arčiau, o skirtingų klasių pavyzdžiai toliau [13].

Šioje vietoje svarbi viena problema. K geriausių kandidatų vertinimo rodikliai priklauso nuo visos kandidatų eilės, todėl jų sunku mokyti per atskiras poras. Be to, porų ir trijų pavyzdžių grupių labai greitai daugėja, todėl praktikoje tenka rinktis tik dalį sunkesnių pavyzdžių. Dėl to gali keistis ir tai, ko modelis iš tikro mokosi. Todėl literatūroje atsiranda mokymas pagal visą kandidatų sąrašą. Jei galutinis tikslas yra geras rikiavimas, prasminga mokymą kuo labiau priartinti prie rodiklio, kuris vertina visą kandidatų eilę, o ne tik atskiras poras[14].

Mokymas pagal visą kandidatų sąrašą literatūroje siūlomas tam, kad būtų sprendžiama tokia problema. Vertinimas šiame darbe remiasi rikiavimu, bet mokymas paprastai nėra tiesiogiai taikomas prie rikiavimo rodiklių. Dėl to siūlomi nuostoliai, kurie labiau priartina mokymą prie viso kandidatų sąrašo tvarkos [8, 14]. Vis dėlto toks kelias reiškia kitokių požymių vektorių mokymą ir didesnę nustatymų paiešką, todėl sąrašiniai nuostoliai tiesiogiai nebuvo taikyti. Kaip artima idėja buvo patikrintas FastAP, tačiau jis rezultatų nepagerino (žr. 1 priedo 1.2 poskyrį).

FastAP gerai parodo tą pačią problemą. Straipsnyje požymių vektorių mokymas aprašomas kaip mokymas kuo geriau surikiuoti kandidatų sąrašą. Siūloma mokymo metu tiesiogiai taikytis į vidutinį tikslumą (angl. average precision, AP), o pats skaičiavimas supaprastinamas taip, kad jį būtų galima praktiškai panaudoti. Jei galutinis tikslas yra geras kandidatų rikiavimas, mokymo nuostolio funkcija turėtų būti kuo artimesnė rikiavimo metrikai, o ne tik netiesiogiai remtis atstumais ir atskyrimo riba

(angl. margin). Tai parodo, kad požymių vektorių mokymo pasirinkimas, nuostolio funkcija ir mini aibių (angl. mini-batch) sudarymo būdas yra ne vien įgyvendinimo detalė, o sprendimą lemiantis dalykas [8]. FastAP šiame darbe patikrintas vienu bandomuoju variantu, tačiau testavimo makro F1 buvo 0,7335 ir liko žemiau bazės, todėl jis paliktas kaip tolesnė kryptis, kuriai reikėtų daugiau nustatymų paieškos (žr. 1 priedo 1.2 poskyrį).

FastAP šiame darbe svarbus todėl, kad jis bando mokymą priartinti prie to, kaip vėliau iš tikro vertinamas kandidatų sąrašas. Monetų atveju tai svarbu kaip argumentas, kad nuostolio parinkimas turi būti siejamas su tuo, kaip realiai skaičiuojamas rikiavimas ir kaip vertinami K geriausi kandidatai, o ne tik su tuo, ar pavyko atskirti poras treniruotėje.

Praktikoje monetų duomenyse papildomas iššūkis yra žymėjimo (etiketės) nuoseklumas ir detalumo lygis. Straipsnyje apie kontekstinį panašumą pastebima, kad daugelis metrinio mokymo nuostolių gali per stipriai prisitaikyti prie semantiškai nenuoseklių etikečių (perdėtas prisitaikymas, angl. overfitting), o tai ypač aktualu, kai reali semantika yra hierarchinė arba kai teisingo atitiktens sąvoka priklauso nuo vertinimo scenarijaus [15]. Monetų atveju analogiška situacija atsiranda tada, kai vizualiai panašios monetos skiriasi smulkiomis detalėmis. Viename vertinime jos būtų pakankamai panašios, pavyzdžiui, iki vertės lygio, o kitame tai jau būtų klaida, pavyzdžiui, iki monetos leidybos tipo. Todėl ši literatūra padeda pagrįsti metodinį sprendimą. Prieš parenkant nuostolį ir metrikas, reikia aiškiai apibrėžti, kokį detalumo lygį laikome tiksline atitiktimi ir ar reikia papildomų grupių ar hierarchinio vertinimo.

Svarbu, kad šiame darbe kontekstinis panašumas (angl. contextual similarity) suprantamas kaip dviejų pavyzdžių kaimynų persidengimo dalis, tai yra kiek artimiausių kaimynų jie turi bendrų, ir pažymima, kad tokia korekcija dažnai naudojama vertinimo metu, tačiau standartiniai mokymo nuostoliai jos tiesiogiai neoptimizuoja. Autoriai siūlo konstruoti kontekstinį nuostolį ir eksperimentais parodo, kad galima pasiekti didesnę testinio rinkinio atkūrimą@1 (angl. Recall@1, R@1) net esant mažesniai mokymo atkūrimui@1. Tai interpretuojama kaip mažesnis perdėtas prisitaikymas prie triukšmingų ar semantiškai nenuoseklių etikečių. Papildomai parodomas pranašumas esant dirbtiniam etikečių triukšmui. Monetų identifikavimo kontekste tai suteikia konkrečią alternatyvą vien kosinusiniu panašumu paremtam reitingui. Jei matome, kad klaidos kyla dėl žymėjimo detalumo, pavyzdžiui, kai atskyrimas pagal vertę susimaišo su atskyrimu pagal monetos tipą, kontekstinis perrikiavimas ar kontekstinės optimizacijos idėja yra pagrįsta kryptis, bet ji reikalauja aiškiai apibrėžto kaimynų grafo ir papildomo skaičiavimo [15].

Kontekstinio panašumo literatūra šiame darbe taikyta kaip idėja, kad kandidatą galima vertinti ne tik pagal tiesioginį panašumą su užklausa, bet ir pagal tai, ar jų artimiausių kaimynų aplinka yra panaši. Ši idėja buvo patikrinta paprastu bandomuoju variantu paieškos stadijoje, be papildomo mokymo. Kandidatų sąrašė tvarka buvo keičiama pagal tai, kiek sutampa užklauskos ir kandidato kaimynai, o po to taikytas tas pats papildomos informacijos perrikiavimas. Pagerėjimo tai nedavė. Testavimo makro F1 sumažėjo iki 0,7894, palyginti su baze 0,8190, nors atkūrimas@5 liko panašus ir siekė 0,9761. Todėl kontekstinė idėja liko tik kaip patikra, o ne kaip pagrindinis sprendimas [15].

Atstovų metodų kontekste svarbus klausimas, ar atstovas tikrai atitinka realios klasės pasiskirstymą. Monetų atveju tai tiesiogiai siejasi su praktine problema. Vienos klasės vaizdai gali labai skirtis dėl apšvietimo, nusidėvėjimo ar fotografavimo kampo, todėl vienas centras gali per silpnai aprašyti visą klasę. Šiame darbe toks kelias patikrintas supaprastintu atstovais paremtu nuostoliu (ProxyNCA). Gautas testavimo makro F1 buvo 0,7572, tai yra prasčiau nei bazė 0,8190, todėl pilnas proxy kalibravimo mechanizmas toliau nebuvo plėtojamas (žr. 1 priedo 1.2 poskyrį). Straipsnyje apie atstovų kalibravimą teigiama, kad atstovo ir realios klasės centro nuokrypis gali būti reikšmingas,

todėl siūloma atstovų kalibravimo struktūra (angl. Calibrate Proxy), kuri naudoja realių pavyzdžių informaciją, įveda papildomą kalibravimo nuostolį ir leidžia vienai klasei naudoti kelis atstovus [16].

Hierarchinio atstovų metodo straipsnyje siūloma atstovų hierarchija, kuri leidžia ne tik atskirti klases, bet ir išlaikyti ryšius tarp panašių klasių, ypač dideliuose rinkiniuose. Monetoms tai aktualu todėl, kad panašūs ryšiai tarp klasių yra ir pačiose monetose, pavyzdžiui, vertė ar valdovas apjungia kelis monetos leidybos tipus, tačiau ši informacija anotacijose gali būti nepilna. Dėl to toks metodas yra pagrįsta alternatyva, jei norime, kad artimos klasės požymių vektorių erdvėje liktų arti viena kitos, bet kartu būtų atskirtos tiek, kiek reikia šiam darbui. Hierarchinio atstovų metodo (HPL) šiame darbe neįgyvendinome, nes tam reikėtų patikimos klasių hierarchijos arba papildomo klasterizavimo ir validacijos sluoksnio. Vietoje to klasių sistemos etiketės smulkumo poveikis patikrintas vėliau, sujungiant etiketes į stambesnes grupes.

Netolygiame scenarijuje pasirinkimas, kaip mokyti požymių vektorius pagal panašumą, susikerta su retų klasių problema. PaCo straipsnyje [7] parodoma, kad prižiūrimas mokymas, kuris traukia panašius ir stumia skirtingus gali būti iškraipytas dažnoms klasėms, ir siūlo naudoti klasių vidurio taškus, kad mokymas mažiau nukryptų į dažnas klases. Ši kryptis šiam darbui svarbi kaip realus kandidatas požymių vektorių mokymo modifikacijai, jei eksperimentinė dalis parodys, kad retos klasės sistemingai nepatenka į K geriausių kandidatų net ir po perrikiavimo.

PaCo darbe ši atsvara įvardijama kaip mokomi klasių vidurio taškai, kurie veikia kaip papildomi mokymo taškai greta pavyzdžių tarpusavio traukos ir atstūmimo. Taip siekiama, kad retų klasių pavyzdžiai turėtų stabilesnę tikslą, o ne būtų nustumiami į šalį dėl gausios dažnų klasių porų statistikos. Monetų atveju klasių vidurio taškai gali padėti tada, kai retos klasės sunkiai susitelkia į savo grupę požymių vektorių erdvėje. Vis dėlto reikia stebėti, ar centrai klasėse su labai mažu pavyzdžių skaičiumi netampa per triukšmingi.

Mastelio klausimui, kuris tampa aktualus didinant pavyzdžių rinkinį ar kandidatų rinkinį, vektorinės paieškos literatūroje siūlomi apytikslės artimiausių kaimynų paieškos indeksai, mažesni vektorių įrašai ir GPU pagreitinoti sprendimai. Johnson, Douze ir Jégou aprašo GPU optimizuotą k mažiausių elementų atranką ir parodo, kad tokį komponentą galima naudoti tiek tiesioginei paieškai, tiek apytikslėms schemoms su mažesniais vektorių įrašais, pasiekiant didelio masto artimiausių kaimynų paiešką (angl. k -nearest neighbors, k NN) milijonų ar milijardų vektorių kolekcijose. Šiam darbui straipsnis svarbus ne dėl skaičių perkėlimo į monetų duomenis. Jis parodo, kad K geriausių kandidatų paieškoje kandidatų rinkinio dydį ir paieškos kainą galima valdyti indeksavimu ir mažesniu vektorių saugojimu. Dėl to aiškiai atskiriama paieškos kaina ir pačių požymių vektorių kokybė. Specializuotas vektorinis indeksas, pavyzdžiui, FAISS, nebuvo naudojamas, nes rinkinio dydis leidžia atlikti paiešką ir be tokio papildomo sluoksnio. Mastelio klausimas vertintas per kandidatų rinkinio dydį ir perrikiavimo kainą.

Projektavimo prasme tai dera su dviejų etapų K geriausių kandidatų sistemos idėja. Pirmą greitai formuojamas kandidatų rinkinys pagal vizualinį panašumą, o antrame etape kandidatai perrikiuojami pagal papildomą signalą ir patikimumo kriterijų. Toks skaidymas leidžia aiškiai valdyti kokybės ir kainos kompromisą. Kandidatų rinkinio didinimas gerina šansą, kad teisingas kandidatas apskritai pateks į svarstomą rinkinį, bet didina skaičiavimo kainą. Perrikiavimas ir patikimumas veikia jau ribotame rinkinyje, todėl jų kaina yra mažesnė ir labiau valdoma. Johnson, Douze ir Jégou darbas padeda parodyti, kad pirmo etapo kaštas yra sisteminis, bet sprendžiamas indeksavimo priemonėmis, o ne vien modelio supaprastinimu [17].

Jei pavyzdžių rinkinys didės, paieška brangs, todėl dalį pavyzdžių gali tekti pašalinti. Toks šalinimas neturi būti daromas akiai. Jį reikia vertinti pagal du dalykus. Pirma, kiek sumažėja skaičiavimo kaina. Antra, kiek pablogėja atpažinimo kokybė. Straipsnyje retinimas aprašomas kaip atskira mažesnė klasifikavimo užduotis. Kiekvienam pavyzdžiui paimama informacija apie jo artimiausius kaimynus, o sprendimas palikti ar pašalinti pavyzdį mokomas pagal kelis žinomus retinimo metodus, pvz., ENN ir Drop3. Dėl to retinimą galima vertinti ne pagal nuojautą, o pagal aiškų kokybės ir kainos palyginimą.

Straipsnis apie pavyzdžių atranką duomenų rinkinyje padeda šį klausimą suformuluoti atsargiau. Jis pagrindžia pačią mintį, kad reikia mokytis ar taisyklėmis spręsti, kuriuos pavyzdžius palikti, o kuriuos pašalinti. Monetų paieškoje prie šios minties prisideda šio darbo netolygumo problema. Retoms klasėms svarbus kiekvienas atskiras variantas, pavyzdžiui, nusidėvėjimas, fotografavimo kampas ar apšvietimas, o dažnoms klasėms dažnai būdinga daugiau per daug panašių pavyzdžių. Todėl, jei retinimas būtų taikomas monetų paieškai, jį tikslinga apibrėžti kaip per daug panašių pavyzdžių šalinimą kartu saugant retas klases ir nemažinant jų įvairovės. Toks kriterijus tiesiogiai sieja mastelio sprendimą su darbo objektu. Taip galima mažinti kainą neaukojant retų klasių, kurios ir taip yra pažeidžiamiausios. Šiame darbe retinimas nėra pagrindinis modulis, nes eksperimentai orientuoti į kokybės ir patikimumo pagrindimą fiksuotame rinkinyje. Vis dėlto jis apibrėžia aiškų būdą, kaip vėliau vertinti mastelio kompromisą plečiant pavyzdžių rinkinį. Paprastas pavyzdžių rinkinio retinimo bandymas su centroidų atranka pablogino rezultatą (žr. 1 priedo 1.5 poskyrį), todėl retinimas paliekamas kaip ateities kryptis, kuriai reikėtų retoms klasėms saugią atrankos taisyklių.

Dažnų klasių perteklius gali pasireikšti ne tik pavyzdžių rinkinio retinime, bet ir pačiame kandidatų rinkinyje. Net ir nieko nešalinant iš pavyzdžių rinkinio, artimiausių kaimynų sąrašo viršuje dažnų klasių pavyzdžiai gali atsirasti per dažnai vien dėl jų gausos. Todėl viena praktiška ir mažos rizikos alternatyva retinimui yra kandidatų sąrašo įvairinimas pagal konkrečią užklausą. Pirma skaičiuojamas didesnis kaimynų sąrašas. Po to formuojamas galutinis kandidatų rinkinys ribojant tos pačios klasės kandidatų skaičių. Ši idėja dera su tikslu neapkarpyti retų klasių, nes ji pirmiausia mažina dažnų klasių perteklių kandidatų sąrašo viršuje, o ne šalina retų klasių pavyzdžius. Šiame darbe toks bandymas parodė, kad įvairinimas gali pagerinti rikiavimo kokybę, ypač vidutinių klasių grupėje, net nekeičiant požymių vektorių mokymo, todėl tai yra prasmingas paieškos etapo papildymas (žr. 1 priedo 1.4 poskyrį).

Apibendrinant požymių vektorių mokymo ir paieškos darbus, matyti, kad gerinimas gali vykti bent trimis lygiais, ir kiekvienas lygis sprendžia kitą problemą. Pirma, mokymo tikslas artinamas prie rikiavimo metrikos, kai kyla įtarimas, kad poriniai ar trijų pavyzdžių nuostoliai per blogai atspindi K geriausių kandidatų reitingą. Antra, rikiavimas papildomas kontekstu, kai etiketės gali būti triukšmingos ar nevienodai pateiktos ir vien kosinusinio panašumo nebepakanka. Trečia, požymių vektorių erdvė tvarkoma per atstovus, kai problema yra didelė klasės vidinė įvairovė arba natūrali hierarchija. Netolygumas veikia visus šiuos lygius, todėl atsiranda ir papildomos korekcijos, tokios kaip PaCo, kai problema yra tai, kad optimizacija nustumia retas klases į šalį. Šiame darbe FastAP, ProxyNCA ir PaCo tipo variantas buvo patikrinti kaip bandomieji variantai, o HPL nebandyta dėl anotacijų hierarchijos trūkumo.

Šioje schemoje svarbu pabrėžti, kad mokymo tikslų keitimas savaime negarantuoja pagerėjimo. FastAP ir PaCo literatūra duoda aiškias kryptis, kaip artinti mokymą prie rikiavimo tikslo arba mažinti polinkį dažnų klasių naudai. Tačiau šių metodų pritaikymas praktiškai reiškia papildomą nustatymų paiešką, jautrumą porų ir sąrašų sudarymui ir didesnį eksperimento netikrumą. Todėl šiame darbe jie laikomi aiškiomis alternatyvomis, kurias verta svarstyti, jei paaikškėja, kad retos klasės sistemingai nepatenka į K geriausių kandidatų jau paieškos etape.

Bandomieji variantai patvirtino šią riziką. FastAP, atstovais paremtas ProxyNCA ir PaCo tipo variantas, kuriame naudoti klasių vidurio taškai, šiame duomenų rinkinyje aiškaus galutinio pagerėjimo nedavė.

Ši klasifikacija yra praktiškai reikalinga šiam darbui, nes ji literatūrą paverčia aiškia sprendimo patikros tvarka. Vietoje to, kad bandytume visas idėjas vienu metu, galime pasirinkti kryptį pagal tai, kurioje sekos vietoje matomas ribojimas. Jei pirmas paieškos žingsnis jau negrąžina teisingų kandidatų retose klasėse, tuomet prasmingiausias yra požymių vektorių mokymo korekcijos, tokios kaip PaCo, atstovų kalibravimo idėjos ar pagal visą kandidatų sąrašą vykdomas rikiavimo optimizavimas. Jei teisingas kandidatas dažnai patenka į kandidatų sąrašą, bet neįtikinamai rikiuojamas, arba klaidos siejasi su etikečių smulkumu, tuomet prasmingesnės yra kontekstinės rikiavimo korekcijos ar mokomas modelis kandidatų tvarkai keisti. Jei kokybė gerėja didinant kandidatų rinkinį, bet sistema tampa per brangi, tuomet literatūriškai pagrįsta pereiti prie kandidatų atrankos ir retinimo.

Apibendrinant, požymių vektorių požiūris monetų identifikavime remiasi lanksčiu pavyzdžių rinkiniu, tačiau jam reikia dviejų aiškių sprendimų. Pirma, reikia aiškiai apibrėžti, ką laikome teisinga atitiktimi ir kaip vertinsime K geriausių kandidatų scenarijų. Antra, reikia pasirinkti tokį požymių vektorių mokymo būdą, kuris ne tik duotų gerą bendrą vidurkį, bet ir nepalikėtų retų klasių paraštėje. Aptarta paieškos ir požymių vektorių mokymo literatūra padėjo pagrįsti, kodėl šiame darbe paieška yra tik pirmas žingsnis, o vien jos nepakanka.

Tokia patikra padeda ir projektavimo sprendimams, nes leidžia aiškiai atskirti, kuriame taške verta didinti sistemos sudėtingumą. Pavyzdžiui, užklauso išplėtimas (angl. query expansion) ar klasių dažnio korekcijos balsavime atrodo kaip greitos paprastos taisyklės, tačiau jos nėra universalios. Jos gali padėti tada, kai požymių vektorių erdvė yra stabili ir vyrauja triukšmo mažinimo efektas, bet gali pakenkti, jei sumaišo ribas tarp labai artimų klasių. Ši rizika patikrinta praktiškai. Paprastas AQE tipo (angl. average query expansion) užklauso išplėtimas pablogino rezultatą, nes testavimo makro F1 sumažėjo iki 0,7903 vietoje 0,8190 (žr. 1 priedo 1.3 poskyrį). Todėl tokio tipo taisyklės laikomos pagalbinėmis alternatyvomis, kurias verta taikyti tik remiantis konkrečia klaidų analize, o ne iš anksto laikyti numatytu pagerinimu.

Šiame darbe požymių vektorių sprendimas įgyvendintas kaip K geriausių kandidatų paieška su valdomu kandidatų rinkiniu. Literatūroje siūlomos požymių vektorių mokymo ir paieškos korekcijos buvo tikrinamos bandomaisiais variantais. FastAP, ProxyNCA ir PaCo tipo nuostoliai, užklauso išplėtimas ir kontekstinis perrikiavimas nedavė stabilaus pagerėjimo (žr. 1 priedo 1.2 poskyrį ir 1 priedo 1.3 poskyrį). Tuo tarpu kandidatų diversifikavimas parodė stabilų pagerėjimą rikiavimo tikslumui (žr. 1 priedo 1.4 poskyrį). Tai svarbi išvada, nes ji rodo, kad ne kiekviena iš pirmo žvilgsnio gera idėja veikia šiame duomenų rinkinyje.

Paieška pagal požymių vektorių čia yra tinkama kryptis, nes ji leidžia grąžinti kelis realius kandidatus, o ne verčia sistemą iškart spėti vieną atsakymą. Tačiau literatūroje siūlytos idėjos nebuvo priimtos aklaui, nes dalis jų su šio darbo duomenimis pagerėjimo nedavė.

1.2 lentelėje apibendrinta, kodėl kandidatų paieška šiame darbe atskiriama nuo kandidatų tvarkos keitimo. Pagrindas paliekamas kelių galimų monetų paieška, o kitos idėjos tikrinamos tik tada, kai jos gali atsakyti į konkretų klausimą.

1.2 lentelė. Požymių vektorių požiūrio ir kandidatų paieškos literatūros suvestinė

Literatūra	Darbo santrauka	Reikšmė šiam darbui
------------	-----------------	---------------------

Romos respublikos monetų paieškos straipsnis [11]	Straipsnyje parodyta, kad dideliame monetų tipų rinkinyje praktiškiau grąžinti kelias galimas monetas negu iš karto rinktis vieną atsakymą. Jame taip pat aptariama, kad reikia iš anksto nuspręsti, kiek kandidatų verta parodyti žmogui.	Dėl to šiame darbe pirmiausia grąžinamos kelios galimos monetos, o ne vienas galutinis atsakymas.
Straipsnis apie mokymą pagal visą kandidatų sąrašą [8]	Straipsnyje siūloma modelį mokyti taip, kad jis geriau sudėliotų visą kandidatų sąrašą, o ne tik lygintų du pavienius pavyzdžius. Taip mokymas daromas artimesnis tikram tikslui, kai svarbu, kokia tvarka pateikiami kandidatai.	Ši idėja parodė, kad kandidatų tvarką būtų galima gerinti jau mokymo metu. Ji nebuvo pasirinkta pagrindiniam sprendimui, nes reikėtų iš naujo ruošti mokymo poras arba sąrašus, parinkti kitą nuostolį ir iš naujo ieškoti nustatymų. Supaprastintas FastAP bandymas buvo atliktas, bet rezultatas sumažėjo iki 0,7335 makro F1, todėl šis kelias nebuvo plėtojamas toliau.
FastAP straipsnis [8]	Straipsnyje siūloma modelį mokyti pagal tai, ar teisingi kandidatai atsiduria aukščiau sąrašė. Taip mokymo tikslas tampa panašesnis į vėlesnį kandidatų rikiavimo vertinimą.	Ši idėja šiame darbe buvo išbandyta, bet galutinio rezultato nepagerino.
Straipsnis apie kontekstinį panašumą [15]	Straipsnyje parodyta, kad vien pradinio panašumo gali nepakakti, kai žymos netikslios arba per smulkios. Siūloma lyginti ir tai, ar užklausa ir kandidatas turi panašius artimiausius kaimynus.	Ši idėja buvo išbandyta kandidatų tvarkai tikslinti, bet galutinio rezultato nepagerino.
Straipsnis apie klasės atstovų tikslinimą [16]	Straipsnyje aiškinama, kad vienas klasės atstovas gali per blogai atspindėti visą klasę. Atstovas čia reiškia mokymo metu sukurtą klasės centrą, su kuriuo lyginami nauji vaizdai. Jei klasė labai įvairi, vieno centro gali nepakakti, todėl straipsnyje siūloma naudoti kelis atstovus.	Ši idėja buvo patikrinta supaprastintu ProxyNCA bandymu. Rezultatas buvo 0,7572 makro F1, mažiau negu pagrindinio kNN varianto 0,8190. Dėl to sudėtingesnis kelių atstovų ir kalibravimo kelias šiame darbe nebuvo plėtojamas.
Straipsnis apie kelių lygių klasių grupavimą [5]	Straipsnyje siūloma klases grupuoti keliais lygiais, kad modelis matytų ne tik vieną konkrečią klasę, bet ir platesnę jos grupę. Toks kelias gali padėti tada, kai giningos klasės sudaro aiškias grupes.	Šis kelias šiame darbe netaikytas, nes monetų klasės čia neturi iš anksto paruošto ir patikimo grupavimo.
PaCo straipsnis [7]	Straipsnyje parodyta, kad požymių vektorių mokymas gali labiau padėti dažnoms klasėms, todėl siūloma mokymo metu naudoti kiekvienos klasės centrą. Taip siekiama, kad retos klasės nuo pat pradžių neatsiliktų.	Šis metodas šiame darbe buvo patikrintas PaCo tipo bandymu. Jis pasiekė 0,7656 makro F1 ir nepagerino pagrindinio kNN varianto. Todėl PaCo liko kaip galimas tolesnis darbas, jei ateityje būtų aiškiau matyti, kad retos klasės silpnos jau pirmame paieškos žingsnyje.
Straipsnis apie greitą vektorinę paiešką [17]	Straipsnyje parodyta, kad dideliame požymių vektorių rinkinyje nereikia	Šiame darbe kandidatai ieškomi tarp mokymo aibės pavyzdžių. Atskira

	kiekvieną kartą lyginti užklauso su visais įrašais paprastu būdu. Paiešką galima spartinti indeksais, GPU skaičiavimu ir mažesniu vektorių saugojimu, kad būtų greitai randami artimiausi kandidatai.	palyginimo bazė, kuri būtų atskirta nuo mokymo aibės ir naudojama tik kandidatams rasti, nebuvo sudaryta. Duomenų rinkinys dar nėra toks didelis, kad reikėtų FAISS ar kito specialaus indekso. Todėl pirmiausia vertinama, ar požymių vektoriai duoda gerus kandidatus. Greitesnis indeksavimas paliekamas ateičiai, jei pavyzdžių rinkinys padidėtų ir paprastas palyginimas su visais įrašais taptų per lėtas.
Straipsnis apie pavyzdžių atranką duomenų rinkinyje [18]	Straipsnyje pavyzdžių atranka aprašoma kaip sprendimas, kuriuos pavyzdžius palikti, o kuriuos pašalinti. Ši mintis čia naudojama atsargiai. Mažinant paieškos rinkinį reikia tikrinti ne tik greitį, bet ir tai, ar nelieka per mažai naudingų pavyzdžių retesnėms klasėms.	Ši mintis liko svarbi kaip būsimas kelias, jei pavyzdžių rinkinys didėtų. Šiame darbe paprastas retinimo bandymas pagerėjimo nedavė.

1.3. Papildoma informacija ir kandidatų perrikiavimas

K geriausių kandidatų paieška pagal požymių vektorius sumažina uždavinį iki kandidatų sąrašo, tačiau monetų domeno literatūra rodo, kad ir tada dažnai lieka keli labai panašūs galimi atsakymai. Vizualiai panašios monetos gali priklausyti skirtingoms klasėms, ypač kai skiriasi smulkūs laikotarpio ar kiti struktūriniai požymiai. Dėl to vien vaizdo nepakanka, o metai, svoris, skersmuo ar kiti monetos laukai tampa antru informacijos šaltiniu.

Vienas aiškus monetų domeno pavyzdys yra sprendimas, kuris jungia pavyzdžiais paremtą vaizdo atitikimą (angl. exemplar-based matching) su monetos teksto atpažinimu. Kai vien vaizdo nepakanka, monetos tekstas padeda atskirti labai panašius kandidatus, o galutinis sprendimas gaunamas sujungus kelių modulių balus. Tas pats šaltinis mini ir išankstinę atranką pagal monetų matavimus, pavyzdžiui, skersmenį ir storį (angl. coin measurements), kuri parodo, kad papildomas signalas gali būti naudojamas ir paieškos erdvei sumažinti, tačiau tik tada, kai tas signalas yra pakankamai patikimas ir vienodai pateiktas [19].

Svarbu ir tai, kad monetų domeno sprendimai beveik visada remiasi dviem vaizdais, tai yra priekinės ir galinės pusės vaizdais, bet tai savaime negarantuoja pagerėjimo. Viena pusė gali būti informatyvesnė už kitą, o vaizdų kokybė gali būti nevienoda. Darbe ši prielaida patikrinta komponentų patikra. Palikus tą pačią sistemos seką, bet paieškoje naudojant tik vienos pusės požymių vektorius, testavimo makro F1 sumažėjo nuo 0,8190 (abi pusės) iki 0,7426 (tik priekinė pusė) arba 0,7520 (tik galinė pusė), nors atkūrimas@5 išliko aukštas (0,9756). Tai rodo, kad priekinės ir galinės pusės informacija viena kitą papildo, o dviejų vaizdų naudojimas yra pagrįsta prielaida, kurią verta parodyti eksperimentais, o ne laikyti savaime suprantama (žr. 1 priedo 1.6 poskyrį).

Straipsnyje apie kelių informacijos šaltinių sujungimą monetų atpažinime pateikiamos kelios konkrečios detalės, kurios padeda suformuluoti, kaip papildomas šaltinis turėtų būti įtraukiamas. Eksperimentai atliekami su 232 monetomis iš 60 klasių, kur kiekvienai monetai yra priekinės ir galinės pusės vaizdas, iš viso 464 vaizdai, o galutinis sprendimas gaunamas sujungiant kelių modulių balus. Jame tiesiogiai įvardijama, kad monetos teksto atpažinimo svoriai sujungime parenkami mažesni dėl blogesnės teksto atpažinimo dalies kokybės, t. y. papildomas šaltinis laikomas naudingų, bet neturi vienas nulemti sprendimo. Griežtesni paieškos erdvės apribojimai, pvz., išankstinė atranka pagal skersmens ir storio matavimus, yra prasmingi tik tada, kai papildomas šaltinis yra pakankamai

patikimas ir vienodai pateiktas. Kai patikimumas nevienodas, saugiau pradžioje rinktis negriežtą perrikiavimą be atmetimo [19].

Kitas monetų domeno pavyzdys yra CoinNet tipo sprendimas, kuriame naudojamas požymių sujungimas ir dėmesio mechanizmai (angl. attention) klasifikacijos scenarijuje. CoinNet straipsnis svarbus kaip argumentas, kad papildomo signalo įtraukimas monetų kontekste yra prasmingas, tačiau jis taip pat parodo kainą. Dažnai tam reikia sudėtingesnio kartu mokomo modelio, daugiau nustatymų ir stabilesnių mokymo duomenų. Tai padeda pagrįsti, kodėl šiame darbe papildomas signalas įvedamas atskiru žingsniu, perrikiuojant jau gautą K geriausių kandidatų sąrašą.

CoinNet darbas parodo dar vieną monetoms svarbią priežastį. Vien motyvo atpažinimo gali nepakakti, nes naujose klasėse tas pats pagrindinis motyvas gali atrodyti skirtingai ir turėti skirtingus papildomus simbolius ar tekstus. Straipsnyje pristatomas RRCD rinkinys su daugiau nei 18 000 vaizdų ir 228 galinės pusės motyvų klasėmis. Papildomai atskiriamas RRCD-Main su 100 klasių ir RRCD-Disjoint su 128 klasėmis, kad būtų galima patikrinti, kaip modelis veikia su naujomis klasėmis. Tai rodo, kad monetų atpažinime informacija turi kelis lygius. Motyvo atpažinimas gali sumažinti paieškos erdvę, bet neišsprendžia smulkesnio identifikavimo. Todėl šiame darbe papildomas signalas planuojamas kaip antras etapas po paieškos. Pirma suformuojamas kandidatų sąrašas pagal vizualinį artimumą, o perrikiavimas padeda atskirti smulkesnius atvejus, kai kelios monetos atrodo panašiai. Toks skaidymas leidžia valdyti riziką, kai retoms klasėms trūksta duomenų ir vienas bendras signalų sujungimo modelis tampa mažiau stabilus.

Perrikiavimas (angl. reranking) yra etapas po pirminės paieškos. Iš pradžių sistema sudaro kandidatų sąrašą, o po to pakeičia kandidatų vietas pagal papildomą informaciją ar stipresnį modelį. Šį darbą gali atlikti taisyklė arba mokomas modelis kandidatų tvarkai keisti. Tyrimas apie perrikiavimą kelių žingsnių paieškoje rodo, kad toks žingsnis gali pagerinti atkūrimą@ K , bet kartu pabrėžia, kad tokius modelius reikia mokyti atsargiai, nes jie linkę perdėtai prisitaikyti [20]. Nors tame tyrime nagrinėjamas kitas domenai, jis svarbus kaip atrama perrikiavimo etapui. Tyrimas pagrindžia, kad perrikiavimas yra įprasta kelių žingsnių paieškos sistemos dalis ir kad jo mokymo bei duomenų atrankos klausimai yra labai svarbūs. Dėl šios priežasties mokomas modelis kandidatų tvarkai keisti laikomas papildoma kryptimi, kuriai reikia atskiros validavimo tvarkos.

Praktiniai bandymai patvirtino šią riziką. Net įvedus į retas klases labiau orientuotą užklausų atranką, mokomas modelis kandidatų tvarkai keisti buvo aiškiai prastesnis už paprastą ir paaiškinamą perrikiavimą pagal taisykles. Testavimo makro F1 buvo 0,7119, o bazinis taisyklių variantas pasiekė 0,8190 (žr. 1 priedo 1.7 poskyrį). Tai svarbu todėl, kad sudėtingesnis modelis dar nereikia stabilesnio sprendimo. Kad toks kelias realiai duotų naudą, greičiausiai reikėtų daugiau ir tikslesnės informacijos apie kandidatus arba mokymo, kuris tiesiogiai gerintų viso kandidatų sąrašo tvarką.

Papildomuose bandymuose taip pat tikrintas neuroninis porinis verifikatorius [21]. Tai modelis, kuris kiekvieną užklausos ir kandidato porą vertina atskirai ir bando nuspręsti, ar kandidatas tikrai tinka tai pačiai monetos klasei kaip užklausa. Ši kryptis artima modeliams, kurie lygina du vaizdus tarpusavyje, o ne tik priskiria vaizdą vienai klasei. Šiame darbe toks modelis naudotas kaip antras etapas po paieškos, kai norėta patikrinti, ar papildomas porinis palyginimas gali geriau išskirti teisingą kandidatą į pirmą vietą.

Papildomai buvo tikrintas ir rikiavimo modelis su LightGBM LambdaRank [22]. Tai ne neuroninis tinklas, o iš daugelio sprendimų medžių sudarytas modelis, mokomas perrikiuoti visą vienos užklausos kandidatų sąrašą. Jis vertina ne vieną porą atskirai, o mokosi, kuriuos kandidatus reikia kelti aukščiau, o kuriuos nuleisti žemiau, kai jau turime pradinį vaizdu pagrįstą sąrašą. Šis bandymas

buvo reikalingas tam, kad būtų patikrinta, ar pagal visą kandidatų sąrašą mokomas modelis gali būti stabilesnis už paprastą kandidatų vietų pakoregavimą pagal taisykles.

Kad perrikiavimas pagal papildomą informaciją būtų aprašytas aiškiai ir tiksliai, svarbu įvardyti sujungimo (angl. fusion) lygmenį. Kelių informacijos šaltinių sujungimo apžvalgoje pateikiama tipinė klasių sistema [23]. Ji skiria ankstyvą, vidurinę ir vėlyvą sujungimo būdą. Vėlyvas sprendimo lygmens sujungimas (angl. decision-level) atitinka situaciją, kai skirtingų signalų balai sujungiami po pirminio sprendimo. Šiame darbe tai reiškia, kad naudojamas vizualinis panašumo balas iš požymių vektorių paieškos ir papildomos informacijos atitikimo balas, o galutinis reitingas gaunamas sujungiant šiuos du balus.

Esminė priežastis, kodėl šiame darbe pasirenkamas kandidatų vietų pakoregavimas pagal papildomą informaciją, o ne griežtas atmetimas, yra papildomos informacijos kokybė. Apžvalga apie kelių informacijos šaltinių sujungimą žemos kokybės duomenų sąlygose išskiria tipines problemas [24]. Tai triukšmas, trūkstami duomenys, signalų netolygumas ir kintanti kokybė tarp pavyzdžių. Monetų papildoma informacija praktikoje gali būti nepilna ar netiksli, todėl griežtas atmetimas rizikuoja pašalinti teisingą kandidatą. Kandidatų vietų pakoregavimas leidžia papildomą informaciją panaudoti kaip pagalbinį signalą, bet nepašalina vizualiai tinkamo kandidato iš sąrašo.

Papildomai svarbu, kad skirtingi signalai prieš sujungimą derėtų tarpusavyje (angl. alignment). Multimodal alignment apžvalgoje pabrėžiama, kad nesuderinti signalai gali priversti sistemą remtis klaidinančiais ryšiais [25]. Vėlyvas sujungimas per perrikiavimą čia mažina riziką. Net jei papildomos informacijos balas sukonstruotas netobulai, jis paveikia tik K geriausių kandidatų reitingą, o ne visą požymių vektorių erdvę.

Šiame poskyryje svarbu ne tik pasakyti, kad papildomas signalas padeda, bet ir palyginti skirtingus jo naudojimo būdus pagal riziką ir naudą. Griežtas atmetimas, tai yra filtravimas, atrodo naudingas, nes sumažina paieškos erdvę. Tačiau jis turi didžiausią klaidos kainą. Jei papildoma informacija netiksli ar nepilna, teisingas kandidatas gali būti pašalintas dar prieš rikiavimą, ir tada K geriausių kandidatų jo nebesugrąžina. Perrikiavimas be atmetimo, priešingai, nekeičia kandidatų rinkinio, todėl sumažina šią negrįžtamumo riziką, bet išlaiko galimybę išnaudoti papildomą informaciją tada, kai keli kandidatai atrodo panašiai. Modelis, kuris po paieškos iš naujo sudėlioja kandidatų tvarką, teoriškai gali išmokti sudėtingesnes taisykles nei rankinė paprasta taisyklė, tačiau tyrime apie perrikiavimą kelių žingsnių paieškoje pabrėžiama, kad toks modelis yra jautrus duomenų atrankai ir perdėtam prisitaikymui, todėl tokia kryptis reikalauja labai apgalvoto eksperimento plano. Bendras kartu mokomas sujungimas gali duoti didesnę naudą, bet jis turi ir didžiausią įgyvendinimo kainą. Tada sunkiau atskirti, ar pagerėjo paieška, ar pagerėjo sujungimo modulis, ir reikia stabilesnių duomenų bei daugiau nustatymų.

Šis palyginimas tiesiogiai tampa pasirinkimo argumentu šiam darbui. Jei papildomos informacijos patikimumas nėra garantuotas, pirmas saugus žingsnis yra kandidatų vietų pakoregavimas pagal papildomą informaciją, o ne griežtas atmetimas. Taip sumažinama klaidingo atmetimo rizika ir galima aiškiai pamatuoti, ar papildoma informacija realioje sistemoje iš tiesų duoda naudą. Jei vėliau eksperimentai parodytų, kad šis signalas yra stabilus ir pakankamai informatyvus, tuomet būtų pagrįsta svarstyti mokomą modelį kandidatų tvarkai keisti kaip kitą žingsnį.

Apibendrinant, literatūra leidžia argumentuotai pasirinkti papildomos informacijos perrikiavimą kaip būdą įtraukti papildomą signalą. Monetų domeno šaltiniai rodo, kad papildomas signalas realiai padeda spręsti situacijas, kai keli kandidatai atrodo panašiai. Kelių žingsnių paieškos literatūra pagrindžia perrikiavimo etapą kaip įprastą atkūrimo@K gerinimo būdą. Kelių informacijos šaltinių

sujungimo apžvalgos rodo, kad dėl triukšmo ir nepilnumo saugiau rinktis vėlyvą sujungimą per negriežtą perrikiavimą, o ne griežtą atmetimą.

Kad toks kandidatų vietų pakoregavimas būtų praktiškai stabilus, svarbu aiškiai apibrėžti, kaip skaičiuojamas papildomos informacijos atitikimo balas. Apžvalgoje apie prastos kokybės kelių šaltinių duomenų sujungimą pabrėžiama, kad trūkstami duomenys ir nevienoda kokybė tarp pavyzdžių yra sisteminė problema, todėl sujungimo mechanizmas turi būti atsparus daliniam signalui. Monetų kontekste tai reiškia, kad svorio ir skersmens panašumas turi būti skaičiuojamas tik tada, kai abi reikšmės yra žinomos, o trūkumas turi būti aiškiai matomas sprendime. Panašiai datų atitikimą prasminga apibrėžti kaip intervalų atitikimą su leidžiamu skirtumu, nes datos dažnai būna apytikslės [24].

Papildomai svarbu, kad papildoma informacija būtų suprantama kaip pagalbinis signalas, o ne kaip griežta taisyklė be išimčių, nes praktikoje skirtingos klasės gali turėti labai panašias matmenų ar metų ribas. Dėl šios priežasties kandidatų vietų pakoregavimas yra saugesnis kelias. Jis leidžia papildomai informacijai pastumti reitingą ten, kur vizualiai yra kelios labai panašios alternatyvos, bet neleidžia jai visiškai perimti sprendimo.

Papildomas iššūkis atsiranda tada, kai papildoma informacija yra ne tik skaitiniai ar intervaliniai, bet ir kategoriniai, pavyzdžiui, monetos leidybos vieta, monetų grupė, metalas ar kiti klasių požymiai. Tokie laukai teoriškai gali būti informatyvūs, tačiau praktikoje jie dažnai yra nevienodai koduojami, turi sinonimų, skirtingus užrašymo būdus arba trūksta dalyje įrašų. Dėl to jų įtraukimas turi būti atliekamas kaip papildomas balas, o ne kaip absoliutus sutapimo reikalavimas. Šiame darbe bandomasis monetos leidybos vietos termino įtraukimas kaip mažo svorio perrikiavimo komponentas stabilaus pagerėjimo nesuteikė, todėl pagrindiniame variante kategorinių papildomos informacijos nenaudojame (žr. 1 priedo 1.8 poskyrį) [24].

Papildomos informacijos triukšmas ir nepilnumas lėmė, kad ji naudojama taip, jog tik pakoreguotų kandidatų vietas sąrašė, o ne juos atmestų. Mokomo modelio kandidatų tvarkai keisti kryptis buvo praktiškai patikrinta ir pasirodė nestabili, todėl pagrindiniame variante paliktas aiškiai paaiškinamas perrikiavimas ir komponentų patikros pagal papildomos informacijos dalis bei trūkstumų pogrupius.

Svoris, skersmuo ir metų laikotarpis padeda tada, kai teisingas kandidatas jau yra pradiniame vaizdo paieškos sąrašė. Šie laukai netinka griežtam atmetimui, nes jų gali trūkti arba jie gali būti netikslūs. Todėl galutiniame variante jie tik keičia kandidatų eilę, o ne vieni nusprendžia atsakymą.

1.3 lentelėje apibendrinta, kodėl papildoma informacija naudojama po paieškos, kai jau rastos kelios galimos monetos. Ji padeda tiksliau surikiuoti kandidatus, bet neturi griežtai atmesti atsakymų, nes dalis laukų gali būti nepilni arba triukšmingi.

1.3 lentelė. Papildomos informacijos ir kandidatų perrikiavimo literatūros suvestinė

Literatūra	Darbo santrauka	Reikšmė šiam darbui
Monetų kelių signalų sujungimo straipsnis [19]	Straipsnyje vaizdo palyginimas derinamas su monetos užrašo atpažinimu ir kitais papildomais duomenimis. Jame parodyta, kad papildomi duomenys gali padėti tada, kai vien vaizdo nepakanka, bet jais negalima remtis per griežtai, jei tie duomenys nėra visiškai patikimi.	Dėl to šiame darbe papildoma informacija naudojama tik po vaizdo paieškos. Ji padeda pakeisti jau rastų kandidatų tvarką, bet vien dėl svorio, skersmens ar metų kandidatas nėra iš karto pašalinamas.

CoinNet straipsnis [26]	Straipsnyje vaizdas ir kiti duomenys jungiami viename kartu mokomame modelyje. Tai parodo, kad papildoma informacija monetų atpažinime gali būti naudinga, bet toks sprendimas reikalauja daugiau duomenų, daugiau nustatymų ir sudėtingesnio mokymo.	Dėl to šiame darbe papildoma informacija nebuvo jungiama tiesiai į pagrindinį modelį, o naudota vėlesniame žingsnyje, kai jau turimas kandidatų sąrašas.
Tyrimas apie perrikiavimą kelių žingsnių paieškoje [20]	Tyrime parodyta, kad po pirminės paieškos kandidatus galima sudėlioti iš naujo naudojant papildomus duomenis arba stipresnį modelį. Kartu pabrėžiama, kad toks antras etapas lengvai pradeda per gerai prisitaikyti prie mokymo duomenų.	Tyrimas pagrindė perrikiavimą kaip atskirą etapą po pirminės paieškos. Kartu jis parodė, kodėl mokomą modelį kandidatų tvarkai keisti reikia tikrinti atsargiai. Toks modelis gali išmokti ne bendrą taisyklę, o mokymo rinkinio atsiktinumus, todėl geras mokymo rezultatas dar negarantuoja geresnio testo rezultato.
Kelių informacijos šaltinių sujungimo apžvalga [23]	Apžvalgoje lyginami trys būdai sujungti vaizdą ir kitus duomenis. Galima viską jungti pačioje pradžioje, galima jungti tarpiniuose modelio žingsniuose, arba galima pirma atlikti paiešką pagal vaizdą ir tik tada pridėti kitus duomenis prie galutinio sprendimo.	Iš šios apžvalgos šiame darbe paimtas vėlyvo sujungimo principas. Pirma sudaromas kandidatų sąrašas pagal vaizdą, o tik po to svoris, skersmuo ir metų intervalas naudojami kandidatų tvarkai pakoreguoti. Taip galima aiškiai patikrinti, ar papildoma informacija tikrai pagerina paieškos rezultatą.
Kelių informacijos šaltinių sujungimo žemos kokybės duomenyse apžvalga [24]	Apžvalgoje pabrėžiama, kad papildomi duomenys dažnai būna nepilni, netikslūs ar nevienodos kokybės. Dėl to saugiau juos naudoti kandidatų vietoms pataisyti, o ne griežtai atmesti dalį kandidatų.	Dėl to šiame darbe pasirinktas perrikiavimas be griežto atmetimo.
Apžvalga apie skirtingų signalų derinimą [25]	Apžvalgoje parodyta, kad skirtingi signalai prieš sujungimą turi derėti tarpusavyje, kitaip sistema gali remtis klaidinančiais ryšiais. Rizika mažesnė tada, kai papildomi duomenys veikia tik nedidelį jau atrinktų kandidatų rinkinį.	Tai sustiprino sprendimą papildomą informaciją naudoti tik jau rastų kandidatų tvarkai pakoreguoti.

1.4. Patikimumas, kada sistema neturėtų pateikti atsakymo, ir aprėptis

Net ir turint K geriausių kandidatų sąrašą, praktinis sprendimas turi atsakyti į dar vieną klausimą. Kada sistema neturėtų pateikti atsakymo, jei nėra pakankamai tikra. Tai ypač svarbu netolygiame scenarijuje, kur retoms klasėms dažnai trūksta pavyzdžių, todėl sistema dažniau būna neužtikrinta. SelectiveNet darbe ši problema siejama su atsakymo atranka (angl. selective prediction) ir su situacijomis, kai sistema nusprendžia atsakymo nepateikti (angl. abstention, reject option). Dėl to šiame darbe patikimumas laikomas būtina sprendimo dalimi, o ne papildomu pasirinkimu [27].

Klasikiniame atsakymo su atranka darbe siūloma SelectiveNet idėja. Modelis mokomas taip, kad būtų geriausias pasirinktai daliai atvejų, į kuriuos sistema apskritai atsako. Tai skiriasi nuo paprastesnio kelio, kai sprendimas atsakymo nepateikti taikomas tik po to, parinkus slenkstį pagal pasitikėjimo įvertį.

Paieškos scenarijuje patikimumo klausimas yra kiek kitoks negu klasifikacijoje. Čia sprendimas yra kandidatų sąrašas, todėl pasitikėjimą galima sieti ne su viena tikimybe, o su pačia sąrašo sandara. Pavyzdžiui, svarbu, kiek skiriasi pirmųjų kandidatų balai ir ar viršus atrodo stabilus. Darbe apie tai, kaip vertinti paieškos tikrumą, pabrėžiama, kad paieškos balai dažnai nėra tiesiog paversti tikimybėmis, todėl patikimumas tampa atskiru tyrimo objektu. Tai ir paaiškina, kodėl šiame darbe patikimumas vertinamas pagal aiškia tvarką, o ne vien pagal nuojautą [28].

Kai dauguma klasių turi mažai pavyzdžių, vien pasitikėjimo balu remtis pavojinga. Darbe apie netolygų mokymą su laikinomis modelio žymėmis pabrėžiama, kad patikimumo slenksčio parinkimas yra atskira problema, o modelio pasitikėjimas gali būti nepatikimas, kai pažymėtų duomenų mažai. Net jei sistema nemoko naujų žymių, jos patikimumo slenkstis turi būti tikrinamas atskirai retoms klasėms [29].

Čia grįžtame prie netolygaus vertinimo tvarkos. Jei literatūra rekomenduoja grupes pagal dažnumą, analogiškai atsakymo nepateikimą prasminga vertinti ne tik bendrai, bet ir pagal dažnas, vidutines ir retas klases. Praktinis kriterijus šiame darbe yra toks. Patikimumo mechanizmas turi mažinti klaidingus užtikrintus atsakymus, tačiau neturi paversti retų klasių nematomomis vien todėl, kad joms per dažnai nepateikiamas galutinis atsakymas.

Paieškos sistemoje klaida nėra tik neteisingas pirmas kandidatas. Praktikoje svarbu atskirti bent du atvejus. Pirmas atvejis yra tada, kai teisingas kandidatas nepatenka į kandidatų sąrašą, tai yra paieškos nesėkmė. Antras atvejis yra tada, kai sistema pateikia klaidingą atsakymą su dideliu patikimumo signalu, tai yra patikimumo arba paaiškinimo nesėkmė. SelectiveNet literatūra leidžia šią antrą problemą aprašyti per kompromisą tarp rizikos ir to, kiek užklausų sistema atsako. Jei sistema daliai užklausų atsakymo nepateikia, galima mažinti klaidingų atsakymų riziką, bet svarbu, kad būtų aišku ir pamatuojama, kiek užklausų sistema vis dėlto atsako. Paieškos tikrumo vertinimo literatūra papildomai pagrindžia, kodėl tokia tvarka reikalinga. Paieškos balai gali būti nepaversti į tikimybę, todėl aukštas balas dar negarantuoja patikimumo.

Svarbiausia rizika netolygiame kontekste yra ta, kad atsakymo atranka gali virsti retų klasių ignoravimu. Jei retoms klasėms sistema dažniau nėra tikra, paprastas slenkstis gali per stipriai sumažinti būtent tai, kiek užklausų sistema atsako retoms klasėms. Tada sprendimas popieriuje atrodys tikslesnis, bet praktiškai bus mažiau naudingas ten, kur jo labiausiai reikia. Todėl šiame darbe svarbu du dalykai. Patikimumo mechanizmas turi būti vertinamas kartu su dažnų, vidutinio dažnio ir retų klasių grupėmis, o ne tik bendrai. Taip pat reikia aiškiai parodyti, ar atsakymo atrankos nauda nepasiekiamą vien retų klasių sąskaita.

Atsakymo atrankos ir paieškos tikrumo vertinimo darbai padeda tiksliau suformuluoti, ką praktiškai reiškia sistema, kuri ne visada pateikia atsakymą. Net jei sistema yra uždaro rinkinio, realiame naudojime pasitaiko netipinių užklausų. Tai gali būti neteisingai sužymėtos monetos, naudojamam katalogui nepriklausantys tipai, netinkami ar stipriai apkirpti vaizdai. Kai užklausa tokia netipinė, per didelis užtikrintumas tampa pavojingas. Todėl šiame darbe sprendimas atsakymo nepateikti laikomas ne silpnumu, o būdu valdyti riziką.

Todėl patikimumo projektavimo kriterijus šiame darbe formuluojamas dviem dalimis. Pirmą, patikimumo skaičius turi surikiuoti tipines užklausas pagal sunkumą, kad kompromisas tarp klaidų rizikos ir to, kiek užklausų sistema atsako, būtų valdomas. Antra, patikimumo mechanizmas neturi tapti retų klasių ignoravimo būdu, todėl reikia žiūrėti ir į tai, kiek užklausų sistema atsako dažnų, vidutinio dažnio ir retų klasių grupėse. Kelių modelių rinkiniai ir Bajeso būdai tikrumui vertinti yra alternatyvus kelias, tačiau jiems reikia kelis kartus paleisti modelį ir skirti daugiau skaičiavimo

kainos. Tokia kryptis būtų prasminga tik tada, jei paprastesni sąrašo balai pasirodytų nepakankamai stabilūs šiame duomenų rinkinyje.

Apibendrinant, patikimumas yra būtina K geriausių kandidatų sistemos dalis. Literatūra apie atsakymą su atranka duoda aiškų pagrindą vertinti kompromisą tarp rizikos ir to, kiek užklausų sistema atsako. Literatūra apie paieškos tikrumo vertinimą primena, kad paieškos balai savaime nėra tikimybės, todėl patikimumą reikia vertinti atskirai. Netolygiame scenarijuje papildomai reikia tikrinti, kaip atvejai, kai sistema atsakymo nepateikia, pasiskirsto pagal dažnumo grupes, nes retoms klasėms sistema gali dažniau būti nepakankamai tikra ir jas atmesti.

Praktinė šios literatūros išvada yra ta, kad patikimumo kriterijus neturėtų būti suprantamas kaip vienas universalus slenkstis. Jei paieškos balai nėra paversti į tikimybę, viena alternatyva yra paversti patikimumo skaičių į suprantamą tikimybę, pavyzdžiui, naudojant skirtumą tarp 1 ir 2 vietos balų (angl. margin), ir sprendimą atsakymo nepateikti valdyti tikimybės ribomis. Tokiu atveju patikimumą galima vertinti ne vien vienu tašku, bet ir per rizikos kreivę, per tai, kiek užklausų sistema atsako, ir per plotą po šia kreive (AURC). Taip galima palyginti, ar balų skirtumas iš tiesų surikiuoja užklausas pagal sunkumą.

Praktikoje vien geros rizikos kreivės ir aiškaus rodiklio, kokiais užklausų daliai sistema pateikia atsakymą, neužtenka, jei sistema turi tenkinti minimalų aptarnaujamų atvejų reikalavimą. SelectiveNet darbe akcentuojama, kad neužtenka pasirinkti tikslą, pavyzdžiui, pateikti atsakymą 0,8 visų užklausų daliai. Dar svarbu, kad realiai pasiekta reikšmė būtų artima šiam tikslui, o ne smarkiai svyruotų priklausomai nuo aibės ar grupės. K geriausių kandidatų paieškoje tai reiškia labai paprastą dalyką. Sistema neturi dirbtinai gerinti metrikos vien todėl, kad daliai užklausų atsakymo nepateikia. Todėl patikimumo slenkstis turi būti parenkamas validacijos aibėje pagal aiškiai įvardytą taisyklę, pavyzdžiui, maksimalizuojant tikslinę metriką su sąlyga, kad sistema pateiktų atsakymą bent minimaliai užklausų daliai. Testavimo aibėje tas pats slenkstis taikomas vieną kartą be papildomo priderinimo. Ši procedūra kartu saugo nuo duomenų nutekėjimo ir sukuria patikrinamą rezultatą.

Netolygiame scenarijuje prasminga vertinti ne tik bendrą rodiklį, kiek užklausų sistema atsako, bet ir tai, kiek užklausų ji atsako dažnų, vidutinio dažnio ir retų klasių grupėse. Taip galima tiesiogiai patikrinti, ar patikimumo mechanizmas nepaverčia sistemos patikima tik dažnoms klasėms, o retas klases dažniau palieka be atsakymo.

Šio poskyrio idėjos darbe įgyvendintos taisykle, kuri pagal skirtumą tarp pirmos ir antros vietos balų parodo, kada sistema neturėtų pateikti atsakymo. Pati riba parenkama validacijos aibėje pagal iš anksto nustatytą reikalavimą, kokiais mažiausiai užklausų daliai sistema turi pateikti atsakymą. Papildomi patikimumo tikrinimai, tiesiogiai susiejant tai, kiek užklausų sistema atsako, su dažnų, vidutinio dažnio ir retų klasių grupėmis, pateikti 1 priedo 1.1 poskyryje. Papildomai balų skirtumas buvo paverstas į tikimybę Platt metodu, kad riba būtų suprantama kaip tikimybės riba (žr. 1 priedo 1.11 poskyrį).

Patikimumas šiame darbe vertinamas kaip atskira sprendimo dalis, nes sistema turi mokėti ne tik atsakyti, bet ir laiku sustoti. Riba, nuo kurios sistema neturėtų pateikti atsakymo, turi būti parenkama validacijoje, o jos poveikis tikrinamas ir pagal dažnumo grupes, ne tik bendru vidurkiu.

1.4 lentelėje apibendrinta, kodėl patikimumas šiame darbe vertinamas ne kaip vienas pasitikėjimo skaičius, o kaip rizikos ir atsakymo aprėpties klausimas. Taisyklė turi mažinti klaidingus užtikrintus atsakymus, bet kartu neturi per dažnai palikti retų klasių be atsakymo.

1.4 lentelė. Patikimumo ir atvejų, kai sistema neturėtų pateikti atsakymo, literatūros suvestinė

Literatūra	Darbo santrauka	Reikšmė šiam darbui
SelectiveNet straipsnis [27]	Straipsnyje parodyta, kad sistema gali būti vertinama ne tik pagal tai, kiek atsakymų pateikia teisingai, bet ir pagal tai, kokiai daliai užklausų ji iš viso pasirenka atsakyti. Tai svarbu tada, kai geriau atsakymo nepateikti, negu pateikti klaidingą atsakymą.	Straipsnis pagrindė, kodėl patikimumas šiame darbe vertinamas kaip atskira sprendimo dalis.
Straipsnis apie tai, kaip vertinti paieškos tikrumą [28]	Straipsnyje aiškinama, kad paieškos balai savaime nėra aiški tikimybė. Dėl to patikimumą reikia vertinti ne pagal vieną skaičių, o pagal tai, kaip atrodo kandidatų sąrašo viršus ir kiek aiškiai pirmas kandidatas lenkia kitus.	Dėl to šiame darbe patikimumas siejamas su kandidatų eilės požymiais. Žiūrima, kiek pirmas kandidatas atsiskiria nuo antro ir ar sąrašo viršus atrodo aiškus, o ne tiesiog tikima vienu paieškos balu.
Straipsniai apie patikimumą netolygiame rinkinyje ir vertinimą pagal dažnumo grupes [7, 10]	Vienas straipsnis parodo riziką, kad ta pati patikimumo taisyklė netolygiame rinkinyje gali nevienodai veikti skirtingas klases. Kitas straipsnis pagrindžia, kad rezultatus reikia rodyti ne tik bendru vidurkiu, bet ir atskirai dažnoms, vidutinio dažnio ir retoms klasėms.	Dėl to šiame darbe tikrinama ne tik bendrai, kiek užklausų sistema atsako. Atskirai rodoma ir tai, ar patikimumo taisyklė nepalieka retų klasių be atsakymo dažniau negu kitų klasių.

1.5. Literatūros analizės išvados

Iš literatūros perimtos trys patikros, kurios vėliau tikrinamos eksperimentiškai. Pirmą, netolygiame rinkinyje neužtenka žiūrėti vien bendro vidurkio, nes retos klasės gali likti paslėptos. Antra, dideliame monetų rinkinyje praktiškiau pirma rasti kelis panašius kandidatus, o ne iš karto versti sistemą pateikti vieną atsakymą. Trečia, papildomi atributai ir patikimumo taisyklė turi būti tikrinami atskirai nuo vaizdo paieškos, kad būtų aišku, kuri sprendimo dalis pagerina rezultatą, o kuri tik paslepia klaidą bendrame rezultate.

Požymių vektorių mokymas pagal panašumą vertinamas kaip atsargi alternatyva, o ne kaip numatytasis pasirinkimas. Toks mokymas naudingas tik tada, kai jo tikslas sutampa su vėlesniu kandidatų rikiavimu. Jei etiketės nėra nuoseklios arba klasės apibrėžtos per smulkiai, toks mokymas gali stiprinti skirtumus, kurie paieškai nėra svarbiausi. Atstovų metodai kelia tą pačią riziką, nes vienas klasės atstovas gali blogai aprašyti vidinę klasės įvairovę. Hierarchiniai metodai būtų prasmingi tik turint patikimą klasių hierarchiją. PaCo kryptis lieka tiesioginė alternatyva, jei retų klasių požymių vektoriai bus per silpni ir reikės mažinti dažnų klasių pranašumą.

Papildomų atributų literatūra nukreipė į modulinį perrikiavimą. Monetų domeno darbai rodo, kad kelių informacijos šaltinių sujungimas padeda tada, kai vien vaizdas neatsiskiria panašių kandidatų. Kartu mokomas sujungimas šiame uždavinyje rizikingas, nes papildomi laukai gali būti nepilni, o klaidų priežastį tada sunku atskirti. Kelių žingsnių paieškos darbai rodo, kad perrikiavimo etapas gali gerinti atkūrimą@K, bet jis taip pat priklauso nuo kandidatų sąrašo kokybės ir mokymo duomenų. Dėl to papildomi atributai šiame sprendime įtraukiami po vaizdo paieškos, vėlyvo sujungimo principu [19] [20] [23].

Patikimumo literatūra parodė, kad monetų atpažinimo sistema neturi būti vertinama vien pagal gražintą kandidatų sąrašą. Reikia matuoti ir riziką, kai sistema per daug užtikrintai pateikia klaidingą

atsakymą. SelectiveNet darbas suteikia pagrindą vertinti kompromisą tarp klaidų rizikos ir atsakytų užklausų dalies. Paieškos tikrumo vertinimo darbai primena, kad paieškos balas dar nėra tikimybė. Netolygūs duomenys papildomai reiškia, kad vienas slenkstis gali nevienodai veikti dažnose ir retose klasėse, todėl atsakymo atranka turi būti tikrinama pagal klasių dažnumo grupes.

Literatūroje liko tarpas, kuris tiesiogiai atitinka šio darbo uždavinį. Netolygaus mokymo darbai dažniausiai tiria vieno atsakymo klasifikaciją. Monetų domeno darbai dažnai pateikia kelis kandidatus, bet mažiau nagrinėja, kada sistema turėtų susilaikyti nuo vieno atsakymo. Kelių informacijos šaltinių sujungimo apžvalgos aprašo sujungimo tipus, bet rečiau vertina juos kartu su kandidatų rikiavimu. Todėl neperimamas vienas metodas. Čia sujungiama vaizdo kandidatų paieška, papildomų atributų perrikiavimas ir patikimumo patikra, o visos dalys vertinamos ta pačia tvarka.

Monetų identifikavimui su K geriausių kandidatų sąrašu reikia vertinimo, kuris matuoja ne tik pirmą atsakymą, bet ir tai, ar teisinga moneta apskritai patenka tarp artimiausių kandidatų. Dėl to naudojamos rikiavimo metrikos ir rezultatų skaidymas pagal klasių dažnumą. Papildomi atributai turi padėti atskirti artimus kandidatus, bet negali sugriauti sprendimo, kai dalis laukų nepilni. Todėl pasirinktas ne griežtas filtravimas, o kandidatų vietų pakoregavimas ir atskiras patikimumo sluoksnis.

Literatūra ir duomenų savybės veda prie trijų žingsnių sprendimo logikos. Pirmiausia pagal požymių vektorius ieškoma K geriausių kandidatų. Tada kandidatų eilė koreguojama pagal svorį, skersmenį ir metų intervalą. Galiausiai tikrinama, ar sistema turi pakankamai pagrindo pateikti vieną atsakymą. Ši seka ne tik aprašoma teoriškai, bet ir tikrinama eksperimentais, todėl galima atskirai spręsti, kada verta grįžti prie sudėtingesnių alternatyvų, pavyzdžiui, klasifikatoriaus korekcijų ar mokomo perrikiavimo.

Šis sprendimo pasirinkimas apibrėžia ir darbo ribas. Monetų identifikavimas čia reiškia atitiktens paiešką pagal pasirinktą etikečių apibrėžimą, o ne pilną numizmatinį katalogavimą. Pilnas katalogavimas apimtų, pavyzdžiui, monetos teksto atpažinimą vaizde (angl. OCR), hierarchinės klasių sistemos kūrimą arba platesnį istorinį aiškinimą. Tokios kryptys literatūroje yra galimos ir kai kuriose monetų sistemose taikomos, bet joms reikia kitokios anotacijų struktūros, kitokio vertinimo ir daugiau duomenų paruošimo. Todėl pasirinktas sprendimas, kurį galima vertinti pagal K geriausių kandidatų rezultatus, klasių dažnumo grupes ir atskiras sprendimo dalis.

Modulinis sprendimas pasirinktas dėl duomenų pobūdžio, o ne vien dėl paprastesnio įgyvendinimo. Kelių informacijos šaltinių sujungimo apžvalgos rodo, kad triukšmingi, nepilni ir nevienodos kokybės duomenys apsunkina kartu vykstantį mokymą. Modelis gali išmolti nestabilias sąsajas, o klaidų priežastį tampa sunku atskirti. Monetų rinkinyje ši rizika reali, nes papildomų laukų gali trūkti, o vaizdo kokybę veikia nusidėvėjimas ir fotografavimo sąlygos. Paieškos, perrikiavimo ir patikimumo atskyrimas leidžia patikrinti, ar nauda ateina iš vaizdo paieškos, papildomų atributų ar atsakymo nepateikimo taisyklės. Tas pats skaidymas parodo, ar nauda pasiekia retas klases, o ne tik dažnas.

Literatūros argumentai čia nėra laikomi savaime tinkamais sprendimais. Antro etapo metodai literatūroje aprašomi kaip jautrūs kandidatų generavimo pasiskirstymui, neigiamų pavyzdžių atrankai ir perdėtam prisitaikymui. Todėl literatūroje pasiūlytos idėjos tikrinamos kaip bandomieji variantai, o ne automatiškai perkeliama į galutinį sprendimą. Neigiami bandymų rezultatai irgi svarbūs. Jie parodė, kad smulkaus detalumo monetų scenarijuje užklausos išplėtimas ar kaimynų sutapimas gali sumaišyti artimų klasių ribas ir pabloginti vieno atsakymo kokybę net tada, kai atkūrimas@ K išlieka aukštas. Tai dera su literatūros perspėjimu, kad rikiavimo korekcijos turi remtis klaidų analize ir sistemos tiksliniu detalumo lygiu, o ne būti taikomos visur vienodai [15].

Papildomai patikrintos kelios literatūroje dažnai minimos variacijos. Sujungimas pagal vietas sąrašė (Reciprocal Rank Fusion) teoriškai mažina skalių ir kalibracijos problemą, tačiau smulkaus denominacijų atskyrimo atveju jis panaikino reikalingą balų skirtumo informaciją ir smarkiai pablogino vieno atsakymo kokybę (žr. 1 priedo 1.8 poskyrį). Dviejų pusių požymių vektorių sujungimo svėrimas α reikšme taip pat nedavė stabilaus pagerėjimo (žr. 1 priedo 1.12 poskyrį). Didelio netolygumo korekcija kNN balsavime (inv_sqrt_freq) nepagerino galutinio sprendimo ir pablogino vieno atsakymo kokybę (žr. 1 priedo 1.8 poskyrį). Šie rezultatai rodo, kad bendros paieškos ir rikiavimo idėjos turi būti pritaikomos pagal uždavinio detalumo lygį ir klaidų struktūrą, o ne perkeliamos tiesiogiai.

Kiti bandomieji variantai parodė tą pačią ribą. Griežtas papildomos informacijos atmetimas, kNN balsavimo taisyklių alternatyvos, statistiniai išankstiniai įverčiai ir kandidatų rinkinio mastelio variacijos nedavė stabilaus pagerėjimo šiame duomenų rinkinyje.

Patikimumo ir retų klasių literatūra reikalauja vertinti sistemą keliais pjūviais, o ne viena vidutine reikšme. Todėl šalia makro F1 naudojamas atkūrimas@K dažnoms, vidutinio dažnio ir retoms klasėms bei patikra, ar teisingas kandidatas pateko į kandidatų rinkinį. Tai atskiria dvi skirtingas klaidas. Vienu atveju teisingas kandidatas nerandamas pradiniam sąrašė, kitu atveju jis randamas, bet po perrikiavimo nepatenka į pirmą vietą (žr. 1 priedo 1.1 poskyrį). Patikimumo įverčiai palyginti pagal rizikos santraukas ir atsakytų užklausų dalį, kad būtų matyti, kuris įvertis geriausiai atskiria lengvesnes užklausas nuo rizikingų. Papildomų atributų nepilnumas tikrintas atskiru bandymu, kuriame validavimo duomenyse dirbtinai padidintas trūkstančių laukų kiekis ir stebėta, kaip kinta rezultatai (žr. 1 priedo 1.9 poskyrį). Ta pati problema svarbi kelių pavyzdžių scenarijuje, kai naujos retos klasės atsiranda palaipsniui (žr. 1 priedo 1.10 poskyrį).

Iš literatūroje minimų kryptių, kurios teoriškai galėtų pagerinti rezultatą, bet šiame darbe nebuvo taikytos, svarbiausios yra tos, kurioms reikia daugiau paruoštų duomenų arba kurios gerokai didina sistemos sudėtingumą. Tai yra monetos teksto OCR, hierarchinės klasių sistemos įvedimas, kartu mokomas kelių informacijos šaltinių sujungimas su kryžminiu dėmesiu ir tikrumo įvertinimas kelių modelių rinkiniais ar Bajeso metodais. Šios kryptys svarbios kaip tolimesni darbai, tačiau šiame darbe pirmiausia pasirinkta išnaudoti mažos rizikos, aiškiai atskirai patikrinamas priemones, kurios leidžia įvertinti, ką pagerina paieška, perrikiavimas ir patikimumo patikra. Papildomai atliktas Platt kalibravimo bandymas, kuriuo skirtumas tarp 1 ir 2 vietos balų paverčiamas į tikimybę, kad pirmasis kandidatas yra teisingas. Kalibracija parodė mažą kalibracijos paklaidą (testavimo ECE = 0,0021) ir leido sprendimą atsakymo nepateikti valdyti tikimybės riba. Pasirinkus 0,95 slenkstį, kuris validacijos aibėje parinktas pagal reikalavimą, kokias mažiausiai užklausų daliai sistema turi pateikti atsakymą, testavimo aibėje gauta, kad sistema atsako į 0,8820 dalį užklausų, o atkūrimas@5 siekia 0,9813 (žr. 1 priedo 1.11 poskyrį). Ši patikra reikalinga tam, kad patikimumo sprendimas būtų lengviau paaiškinamas ir lengvai patikrinamas, o ne tam, kad pakeistų pačios sistemos pagrindinę paieškos ar perrikiavimo logiką.

Trumpa visos literatūros analizės suvestinė pateikta 1.5 lentelėje. Ji parodo, kokias idėjas iš literatūros šis darbas perėmė, kurių kryptių atsisakė ir kaip tai virto galutine sprendimo schema.

1.5 lentelė. Literatūros analizės suvestinė

Literatūra	Darbo santrauka	Reikšmė šiam darbui
------------	-----------------	---------------------

<p>Straipsniai apie netolygų duomenų rinkinį [2, 4, 3]</p>	<p>Šie straipsniai parodė, kad realiuose rinkiniuose vienu klasių būna daug, o kitų labai mažai. Jie taip pat paaiškino, kad dėl to retos klasės pradeda atsilikti dar mokymo metu, net jei bendras rezultatas atrodo neblogas.</p>	<p>Dėl to šiame darbe klasių netolygumas laikomas pagrindine problema, o retų klasių elgsena tikrinama atskirai.</p>
<p>Straipsniai apie vertinimą ir dažnumo grupes [1, 9, 10]</p>	<p>Šie straipsniai parodė, kad vieno bendro rodiklio nepakanka. Reikia atskirai rodyti, kaip sistema veikia dažnose, vidutinio dažnio ir retose klasėse, ir aiškiai aprašyti testavimo tvarką.</p>	<p>Dėl to šiame darbe naudojamos dažnumo grupės, atskira retų klasių analizė ir aiški vertinimo tvarka.</p>
<p>Straipsnis apie požymių perkėlimą iš dažnų klasių į retas [4]</p>	<p>Straipsnyje siūloma dalį informacijos iš dažnų klasių perkelti retoms klasėms. Taip retos klasės gauna stipresnį pagrindą dar prieš vertinimą.</p>	<p>Ši idėja buvo svarstyta, bet nebuvo pasirinkta pagrindiniam sprendimui, nes ji labiausiai tinka vieno atsakymo klasifikatoriui. Šiame darbe toks klasifikatorius retoms klasėms veikė silpnai, todėl pagrindinis sprendimas perkeltas į kandidatų paiešką ir perrikiavimą.</p>
<p>Straipsniai apie klasių grupavimą į platesnes grupes [13, 5]</p>	<p>Šiuose straipsniuose siūloma žiūrėti ne tik į vieną labai siaurą klasę, bet ir į platesnę jos grupę. Toks kelias gali padėti tada, kai tarp klasių yra aiški ir iš anksto paruošta tvarka.</p>	<p>Ši kryptis šiame darbe nebuvo taikyta, nes monetų klasėms čia nebuvo paruošto patikimo grupavimo.</p>
<p>LORT straipsnis [6]</p>	<p>Straipsnyje siūloma mokymo metu nevertinti visų neteisingų klasių vienodai griežtai. Taip retos klasės gauna daugiau galimybių priartėti prie teisingo atsakymo.</p>	<p>Tikras LORT šiame darbe nebuvo tikrintas, nes jis nėra paprastas priedas po paieškos. Reikėtų iš naujo mokyti vieno atsakymo klasifikatorių ir atskirai parinkti, kiek iš įprastos 1 reikšmės paliekama teisingai klasei ir kiek paskirstoma kitoms klasėms. Prieduose pateikti FastAP, ProxyNCA ir PaCo tipo bandymai yra artimi tik tuo, kad jie taip pat keičia vaizdo aprašų mokymo tikslą. Jie nėra LORT, nes nekeičia vieno atsakymo klasifikatoriaus mokymo reikšmių taip, kaip siūlo LORT. Po bazinio</p>

		klasifikatoriaus patikros pagrindinis sprendimas buvo sudarytas kaip kandidatų paieška ir vėlesnis perrikiavimas, todėl LORT liko kaip galima klasifikatoriaus kryptis, o ne kaip šio darbo pagrindinis kelias.
PaCo straipsnis [7]	Straipsnyje siūloma mokymo metu remtis ne tik pavieniais pavyzdžiais, bet ir kiekvienos klasės centru. Taip siekiama, kad retos klasės nuo pat pradžių neatsiliktų nuo dažnų.	Šis metodas buvo patikrintas supaprastintu PaCo tipo bandymu. Rezultatas buvo 0,7656 makro F1, mažiau negu pagrindinio kNN varianto 0,8190, todėl ši kryptis nebuvo plėtojama toliau šiame darbe.
Straipsniai apie uždaro rinkinio klasifikaciją ir kelių kandidatų paiešką [12, 11]	Šie straipsniai parodė, kad mažesniuose ir paprastesniuose monetų rinkiniuose gali pakakti vieno klasifikavimo atsakymo, bet sudėtingesniame ir labiau netolygiame scenarijuje praktiškiau grąžinti kelias galimas monetas.	Dėl to šiame darbe pasirinkta kelių galimų monetų paieška, o ne vieno galutinio atsakymo klasifikacija.
Straipsniai apie kandidatų tvarką ir kontekstinį panašumą [8, 15, 20]	Šie straipsniai parodė, kad svarbu ne tik surinkti kandidatus, bet ir teisingai juos išdėstyti sąraše. Jie siūlo modelį mokyti pagal visą kandidatų eilę arba vėliau ją patikslinti žiūrint į platesnį kandidatų kontekstą.	Dėl to šiame darbe kandidatų tvarka vertinama kaip atskiras klausimas, o keli tvarkos gerinimo būdai buvo tikrinti kaip papildomos kryptys.
Straipsniai apie klasės atstovus [13, 16]	Straipsnyje parodyta, kad vieno klasės atstovo gali nepakakti, jei klasė labai įvairi. Dėl to siūloma klasę aprašyti tiksliau.	Ši mintis šiame darbe padėjo įvertinti, kada paprastas klasės atstovas gali būti per silpnas, bet sudėtingesnis variantas galutinėje sistemoje nebuvo taikytas.
Straipsniai apie paieškos greitį ir pavyzdžių rinkinio mažinimą [17, 18]	Šie straipsniai parodė du atskirus klausimus. Dideliame rinkinyje reikia atskirai spręsti paieškos greitį, o mažinant pavyzdžių rinkinį reikia aiškiai nuspręsti, ką palikti ir ką pašalinti.	Dėl to šiame darbe modelio kokybė ir paieškos greitis vertinami atskirai, o paprastas rinkinio mažinimas nelaikomas savaime saugiu.
Straipsniai apie papildomą informaciją tiesiogiai modelyje [21, 26]	Šie straipsniai parodė, kad vaizdą ir kitus duomenis galima jungti viename modelyje. Toks kelias gali	Dėl to šiame darbe papildoma informacija nebuvo jungiama tiesiai į pagrindinį modelį.

	duoti naudos, bet jis reikalauja daugiau duomenų, daugiau nustatymų ir sudėtingesnio mokymo.	
Straipsniai apie perrikiavimą ir atsargų papildomos informacijos sujungimą [20, 23, 24, 25]	Perrikiavimo darbas pagrindžia antrą etapą po pradinės paieškos. Kelių informacijos šaltinių sujungimo apžvalgos papildomai parodo, kodėl triukšmingus, nepilnus ar prastai suderintus signalus reikia jungti atsargiai.	Dėl to šiame darbe papildoma informacija naudojama jau po paieškos, kaip pagalbiniis signalas kandidatų tvarkai pakoreguoti.
Straipsniai apie patikimumą ir kada geriau atsakymo nepateikti [10, 27]	Ši literatūra parodė, kad sistema turi būti vertinama ne tik pagal tikslumą, bet ir pagal tai, kokiai daliai užklausų ji apskritai pasirenka atsakyti. Ji taip pat perspėjo, kad netolygiame rinkinyje taisyklė gali pernelyg dažnai palikti be atsakymo retas klases.	Dėl to šiame darbe patikimumas vertinamas kaip atskira sprendimo dalis ir tikrinamas ne tik bendrai, bet ir pagal dažnumo grupes.

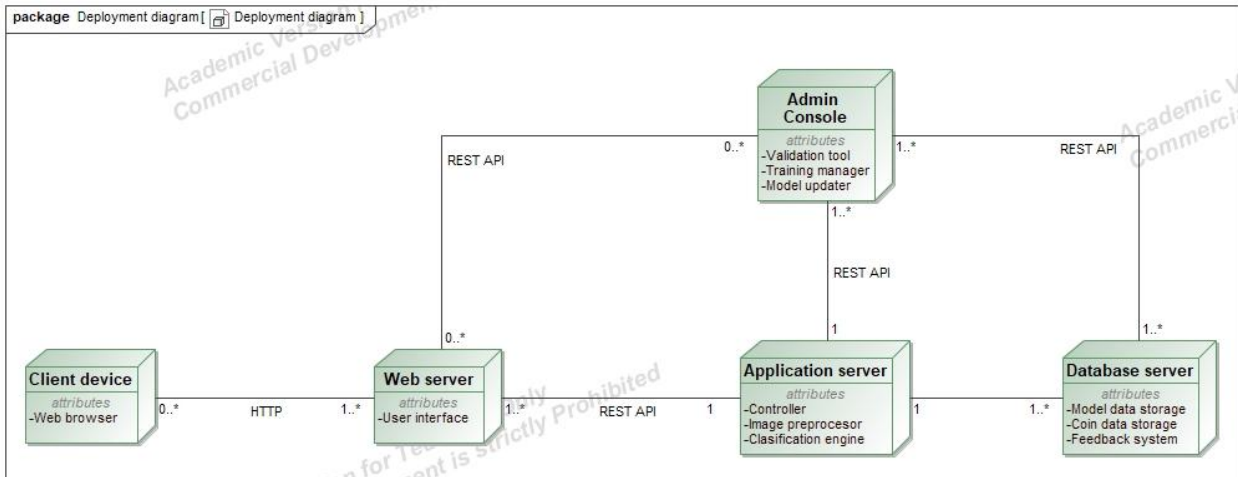
Visa literatūros analizė parodė, kad šiame darbe nepakanka vien bendro tikslumo ir vieno galutinio atsakymo. Netolygaus rinkinio ir vertinimo literatūra parodė, kad retas klases reikia stebėti atskirai. Monetų ir paieškos literatūra parodė, kad šiame scenarijuje praktiškiau pirmiausia grąžinti kelias galimas monetas ir tik po to tvarkyti jų eilę. Papildomos informacijos darbai parodė, kad svorį, skersmenį ir kitus laukus saugiausia naudoti jau po paieškos, kai reikia pakoreguoti kandidatų tvarką. Patikimumo literatūra parodė, kad sistema kartais turėtų geriau nepateikti atsakymo, bet tai reikia vertinti kartu su tuo, kokiai daliai užklausų sistema atsako, ir retų klasių rezultatais.

2. Monetų kandidatų paieškos sistemos projektas

Šiame poskyryje parodyta, kaip sudėta pati sistema ir kaip joje juda duomenys. Diagramos čia reikalingos tam, kad būtų aišku, kur vyksta skaičiavimai, kur saugomi rezultatai ir kaip valdomas modelio atnaujinimas.

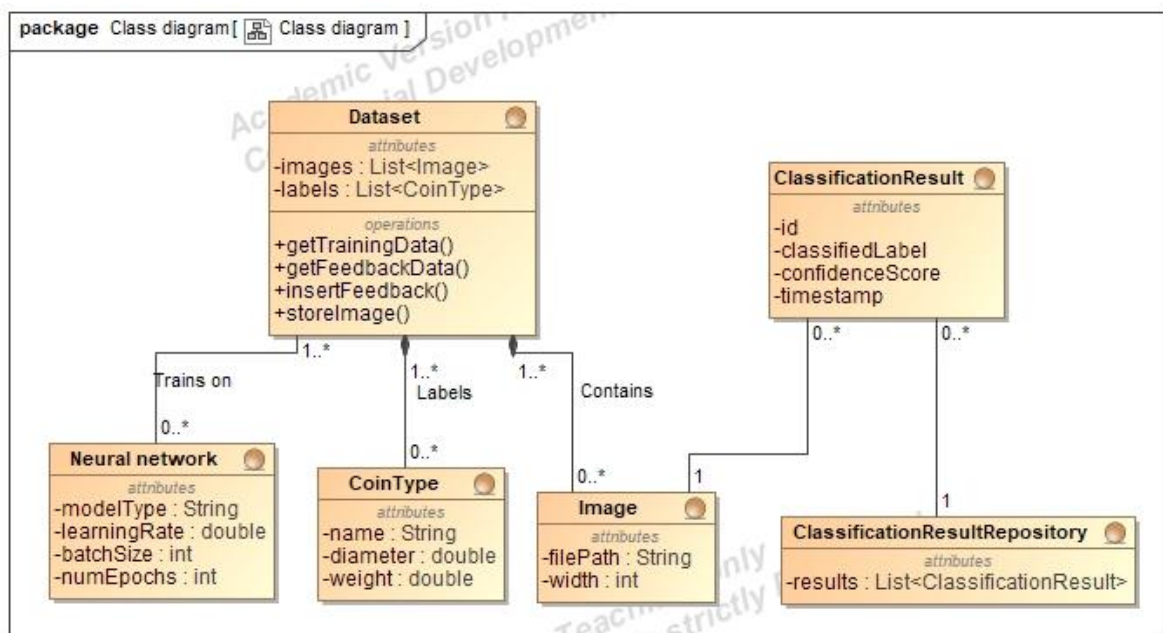
2.1. Statinis sistemos vaizdas

2.1 pav. parodyta sistemos diegimo schema, kurioje naudotojo veiksmai susiejami su serverio skaičiavimais ir duomenų saugojimu.



2.1 pav. Monetų klasifikatoriaus sistemos diegimo diagrama

2.1 pav. parodo pagrindinius komponentus ir jų paskirtį. Naršyklėje naudotojas įkelia vaizdus ir peržiūri gautą rezultatą. Žiniatinklio serveris priima užklausas ir pateikia sąsają, programų serveris atlieka vaizdų paruošimą ir atpažinimą, o duomenų bazė saugo monetų įrašus, modelių versijas ir naudotojų atsiliepimus, kai jie taikomi. Administravimo konsolė reikalinga modelio atnaujinimams ir sistemos būsenos stebėsenai. 2.2 pav. tą patį sprendimą detalizuoja klasių lygiu.

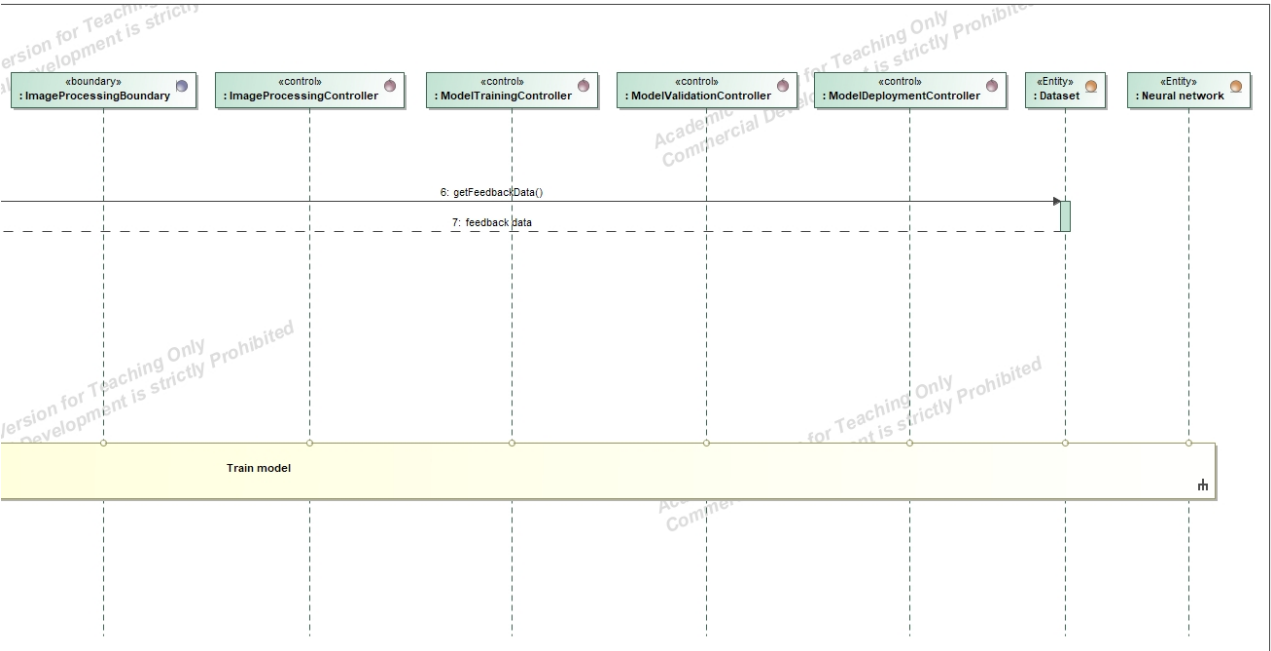
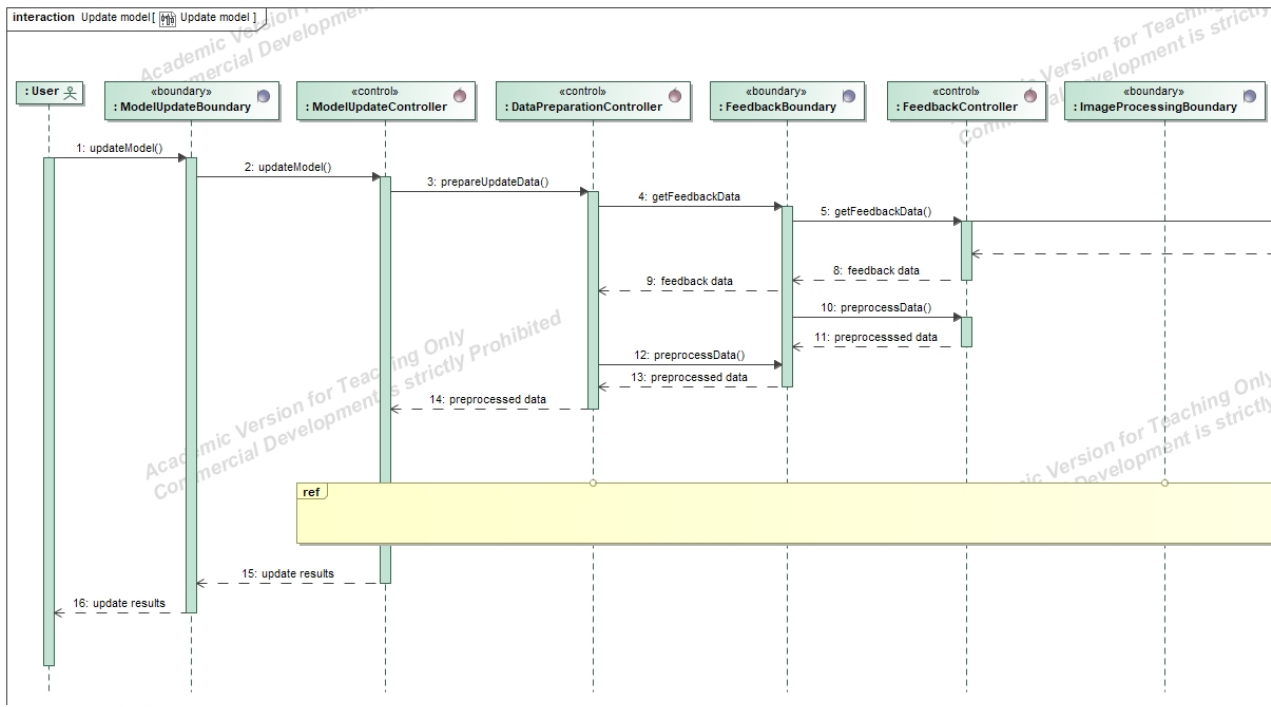


2.2 pav. Monetų klasifikatoriaus sistemos klasių diagrama

2.2 pav. parodo duomenų kelią sistemoje. Duomenų rinkinys pateikia vaizdus su jų žymomis, modelis iš vaizdų sugeneruoja rezultatą, o rezultatai išsaugomi taip, kad vėliau būtų galima atlikti klaidų analizę ir palyginti skirtingas sistemos versijas.

2.2. Dinaminis sistemos vaizdas

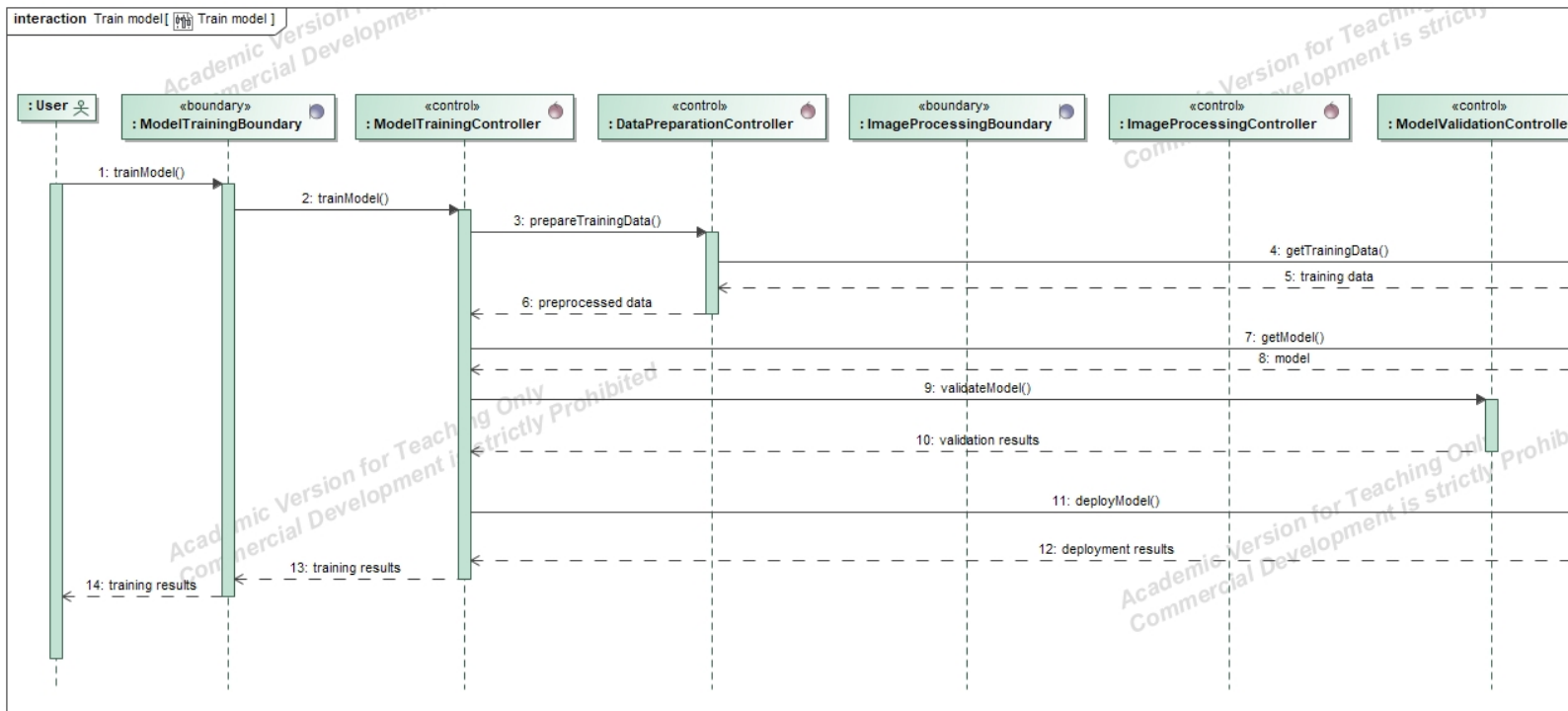
Dinaminės diagramos parodo sistemos veikimą žingsnis po žingsnio. Čia atskiriami du procesų tipai. Pirmasis apima modelio paruošimą ir atnaujinimą, antrasis kasdienį naudojimą, tai yra įkėlimą, atpažinimą ir atsiliėpi



2.3 pav. parodyta modelio atnaujinimo seka. Ji rodo, kad nauja versija diegiama tik po patikros.

2.3 pav. Modelio atnaujinimo sekos diagrama

2.3 pav. rodo atnaujinimo eigą. Administratorius inicijuoja procesą, sistema paruošia duomenis, apdoroja vaizdus, apmoko modelį ir atlieka patikrą. Tik tada nauja versija diegiama ir tampa aktyvi.

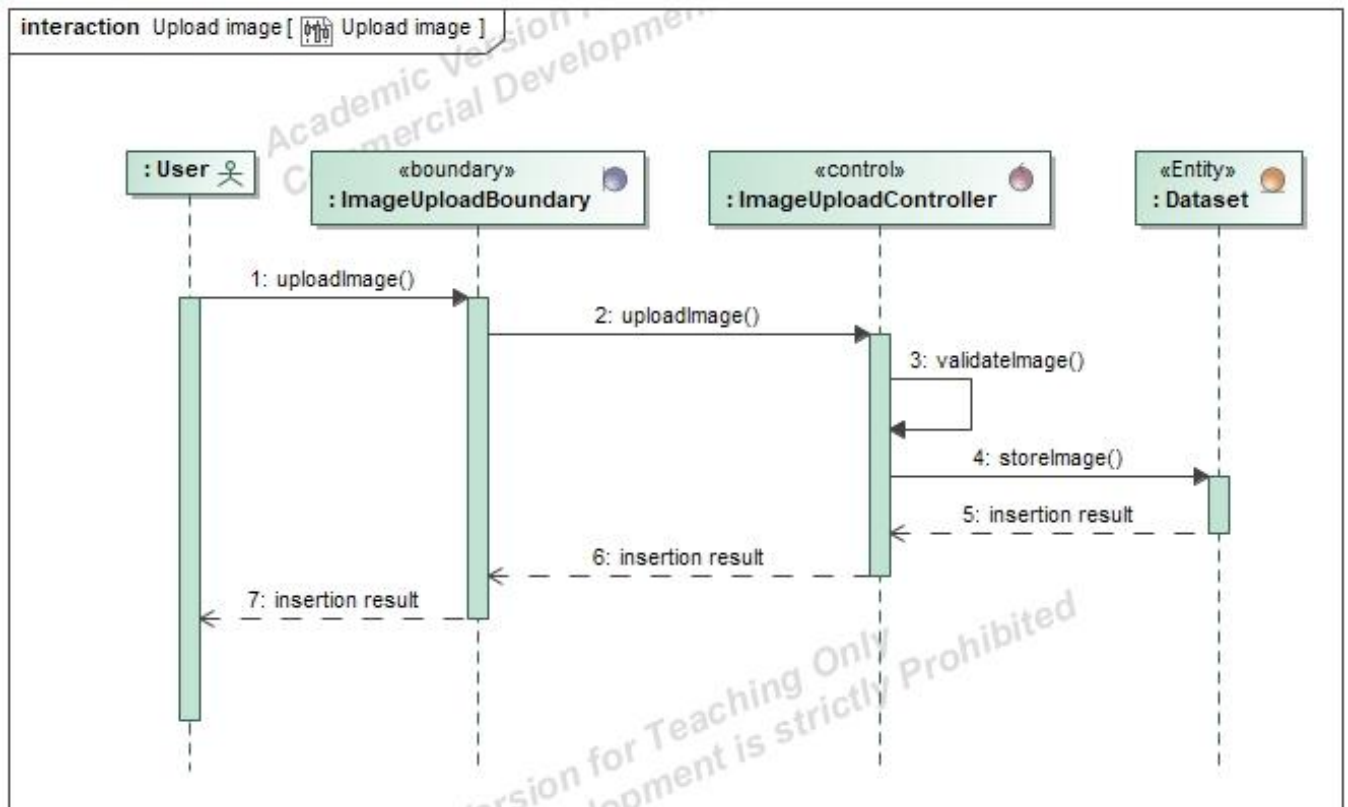


2.4 pav. parodyta mokymo seka, kai modelis mokomas nuo pradžios arba po didesnio duomenų papildymo.

2.4 pav. Modelio mokymo sekos diagrama

2.4 pav. parodo mokymo eigą. Duomenų paruošimas, vaizdų apdorojimas, mokymas, patikra ir diegimas čia sudaro vieną atkartojamą grandinę. Taip sumažėja rizika įdiegti prastesnę modelio versiją.

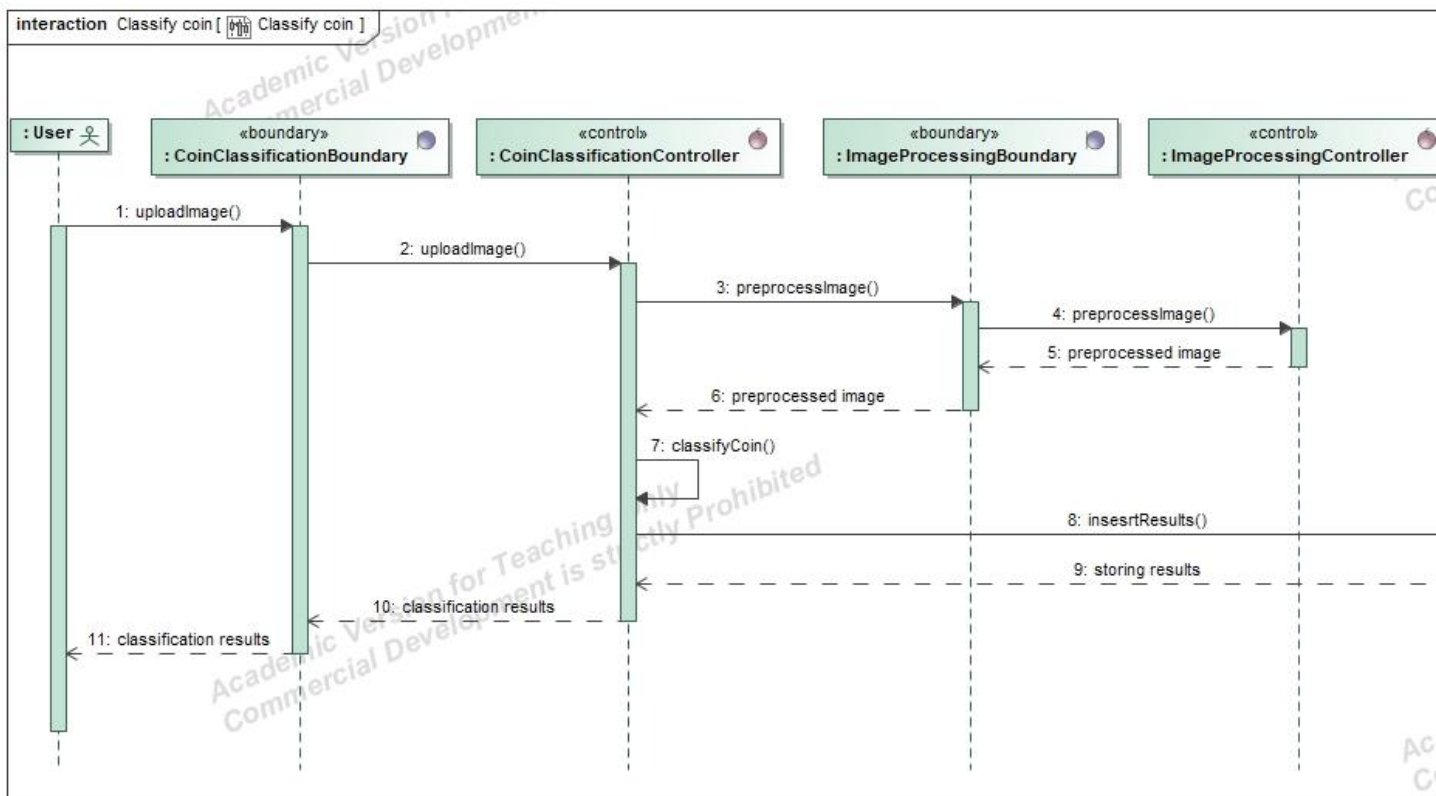
2.5 pav. parodyta vaizdų įkėlimo seka. Ji reikalinga tam, kad į sistemą nepatektų netinkami failai, kurie vėliau sukeltų klaidas.



2.5 pav. Vaizdo įkėlimo sekos diagrama

2.5 pav. rodo, kad įkelti vaizdai pirmiausia patikrinami, tada išsaugomi, o naudotojui grąžinamas aiškus pranešimas apie įkėlimo rezultatą.

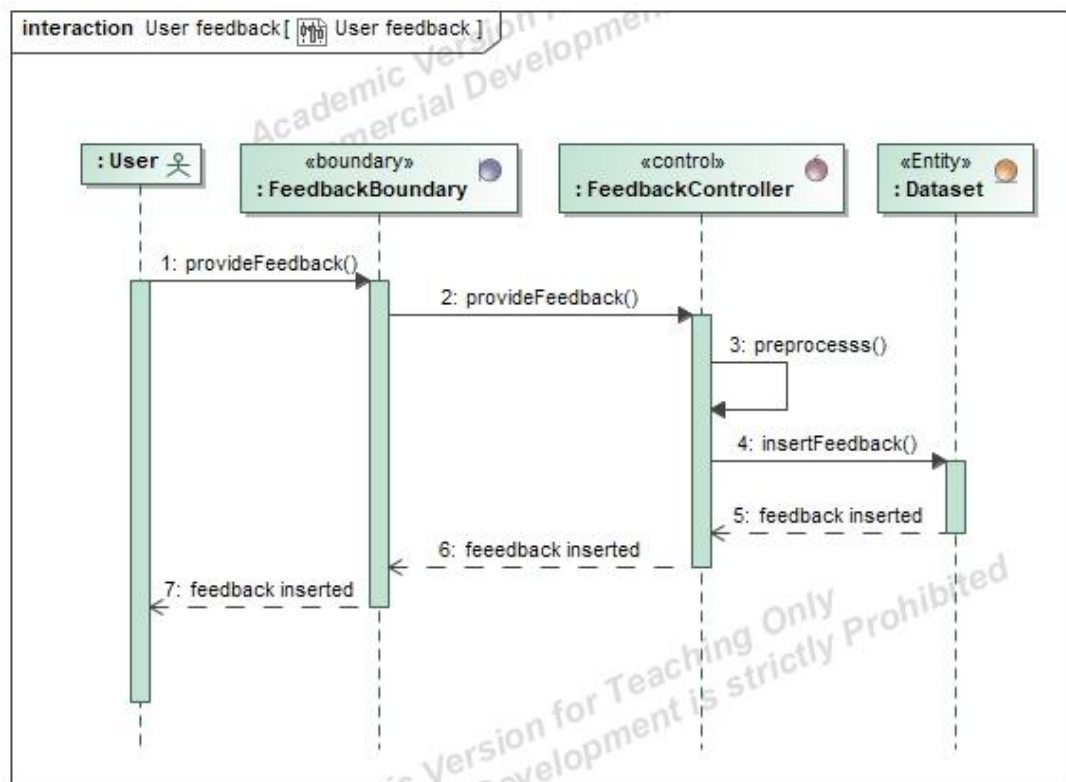
2.6 pav. parodyta pagrindinė monetos atpažinimo seka ir rezultatų įrašymas.



2.6 pav. Monetų klasifikavimo sekos diagrama

2.6 pav. parodo, kad sistema priima vaizdą, jį paruošia ir, naudodama apmokytą modelį, pateikia kelis surikiuotus kandidatus. Rezultatas taip pat įrašomas, kad vėliau būtų galima atlikti klaidų analizę ir stebėti, ar pakeitimai gerina sistemą.

2.7 pav. parodyta, kaip naudotojas pateikia atsiliepimą, kuris padeda identifikuoti tipines klaidas ir planuoti sistemos tobulinimą.



2.7 pav. Naudotojo atsiliepimo pateikimo sekos diagrama

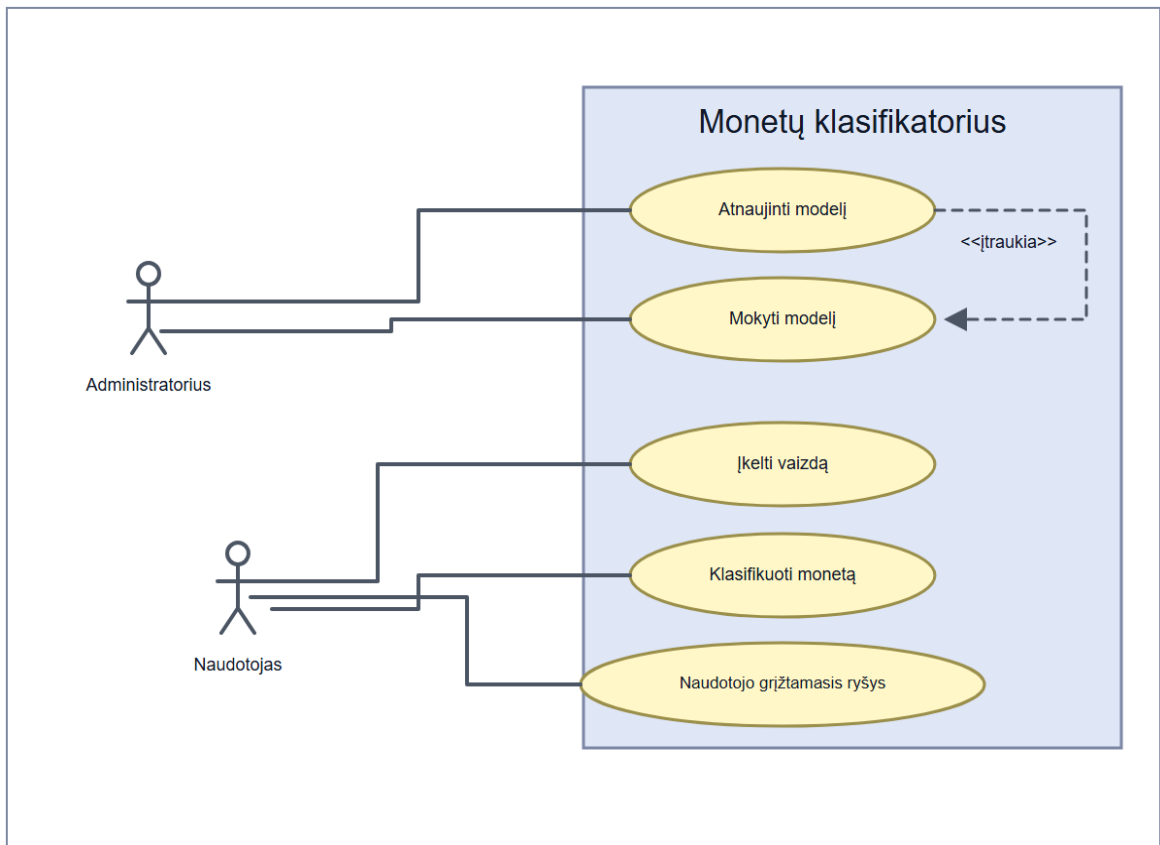
2.7 pav. rodo, kad atsiliepimas priimamas ir įrašomas taip, kad vėliau būtų galima tiksliau suprasti klaidų priežastis ir planuoti duomenų papildymą arba modelio atnaujinimą.

2.3. Komercinė specifikacija

Šį projektą vykdo universiteto studentas Mantas Šutas. Pagrindiniai naudotojai yra monetų kolekcionieriai ir tyrėjai. Projekto pabaigos terminas yra 2026 m. gegužės 16 d. Iki šios datos turi būti paruoštas veikiantis prototipas.

2.4. Sistemos funkcijos

2.8 pav. parodytos pagrindinės sistemos funkcijos dviem naudotojų tipams, tai yra naudotojams ir administratoriams.



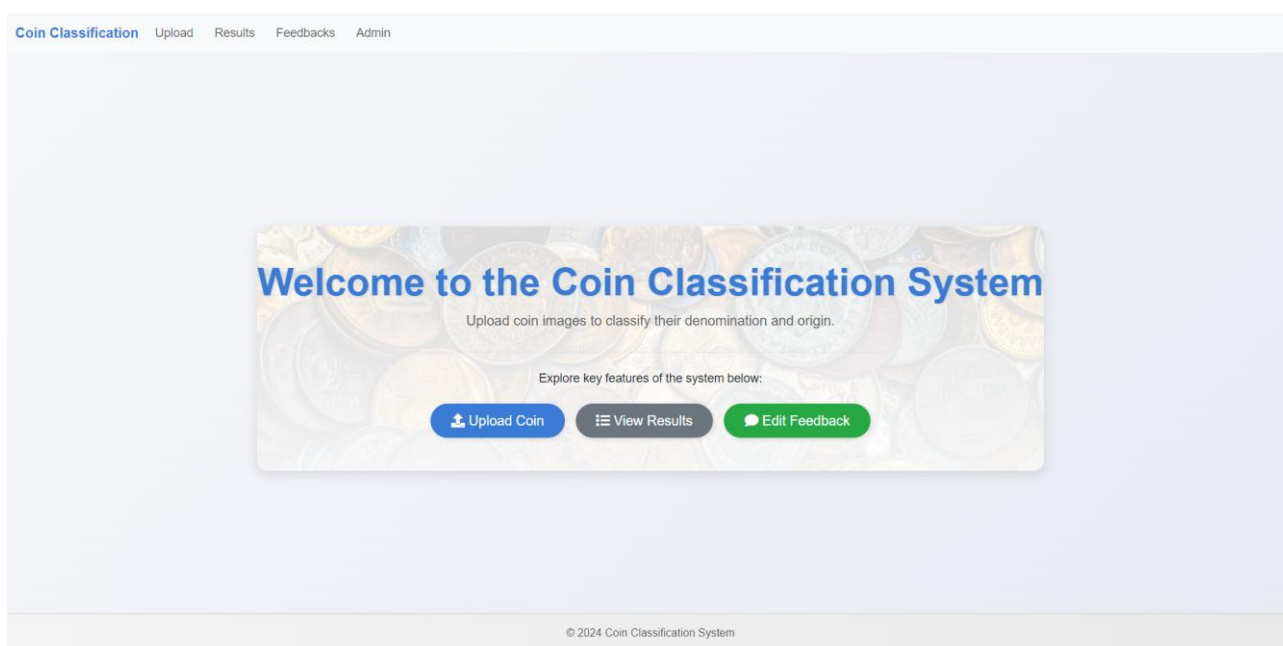
2.8 pav. Monetų klasifikatoriaus sistemos atvejų diagrama

2.8 pav. parodo pagrindines funkcijas. Naudotojas gali įkelti monetos vaizdus, gauti rezultatą, peržiūrėti detales ir pateikti atsiliepimą. Administratorius gali mokyti ir atnaujinti modelį, valdyti jo versijas ir stebėti sistemos veikimą. Toks atskyrimas reikalingas tam, kad kasdienis naudojimas ir modelio priežiūra nesimaišytų.

3. Monetų atpažinimo naudotojo sąsaja

Pagrindinis svetainės puslapis (3.1 pav.) skirtas tam, kad naudotojas iš karto matytų, ką sistema daro ir kur jam eiti toliau. Jame pateikiamas trumpas sistemos paskirties aprašas ir pagrindiniai keliai į svarbiausias dalis.

Puslapio centre pateikti trys pagrindiniai mygtukai. Jie leidžia įkelti monetą klasifikacijai, peržiūrėti klasifikacijos rezultatus ir redaguoti pateiktus atsiliepimus. Taip naudotojas pagrindinius veiksmus pasiekia iš karto, be papildomos paieškos po meniu.



3.1 pav. Pagrindinis puslapis su naudotojo navigacijos pasirinkimais

Įkelimo puslapis (3.2 pav. ir 3.3 pav.) suteikia vartotojams galimybę pridėti naujas monetas sistemai, pateikiant nuotraukas ir papildomą informaciją. Vartotojas privalo įkelti monetos priekio ir galo nuotraukas, o papildomi laukai, tokie kaip svoris, skersmuo, laikotarpis, autoritetas, vertė ir monetos leidybos vieta, yra pasirenkami.

Kai vartotojas užpildo formą ir pateikia duomenis, jie siunčiami į serverį, kur monetos informacija yra apdorojama ir klasifikuojama. Po klasifikavimo vartotojas gali peržiūrėti rezultatus ir, jei reikia, pateikti atsiliepimą apie klasifikacijos kokybę.

Sėkmingo pateikimo atveju, sistema pateikia grįžtamąjį pranešimą. Jeigu aptinkama klaidų, vartotojas informuojamas apie klaidos pobūdį.

The screenshot shows the 'Upload Coin' form with the following fields:

- Obverse Image: Choose File, No file chosen
- Reverse Image: Choose File, No file chosen
- Weight (g): e.g., 5.5
- Diameter (mm): e.g., 24.5
- Start Date: e.g., 1901
- End Date: e.g., 2000
- Authority: e.g., Roman Empire
- Denomination: e.g., Denarius
- Mint: e.g., Rome

© 2024 Coin Classification System

3.2 pav. Monetos įkėlimo forma su papildomų duomenų pateikimu

The screenshot shows the 'Classification Details' modal window with the following content:

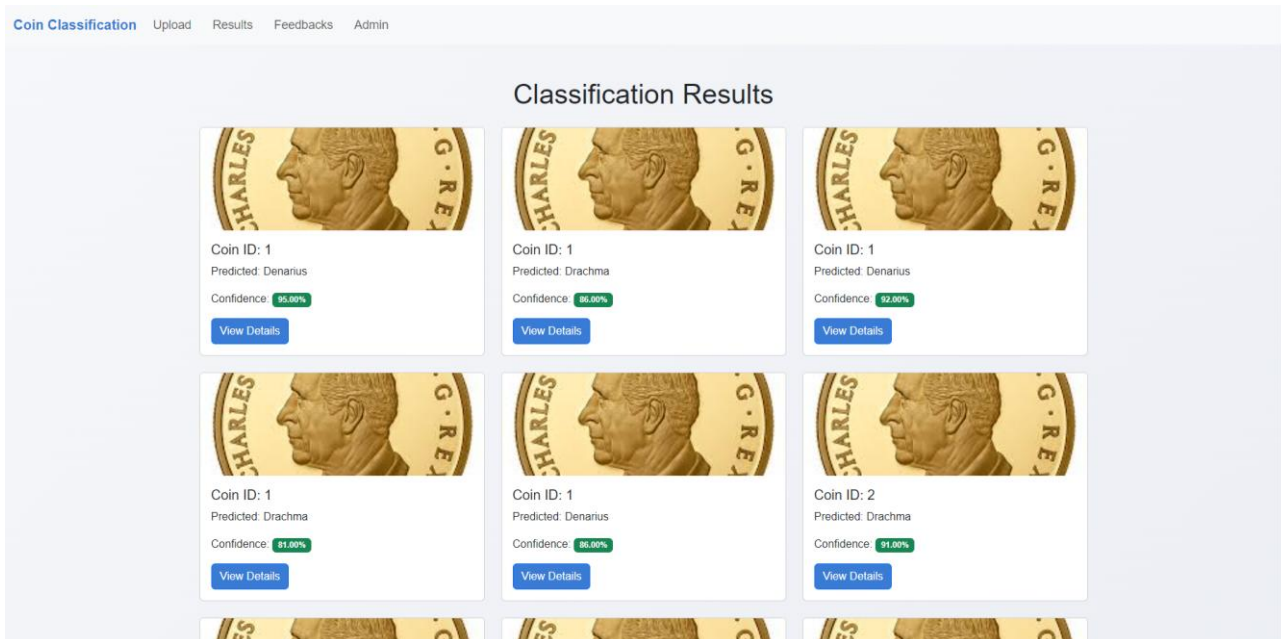
- Obverse Image:
- Reverse Image:
- Classification Details:
 - Predicted Denomination: Sesterius
 - Confidence Score: 84.00%
- Provide Feedback:
 - Correct Label (if incorrect): Enter the correct denomination
 - Comments (optional): Provide additional details

© 2024 Coin Classification System

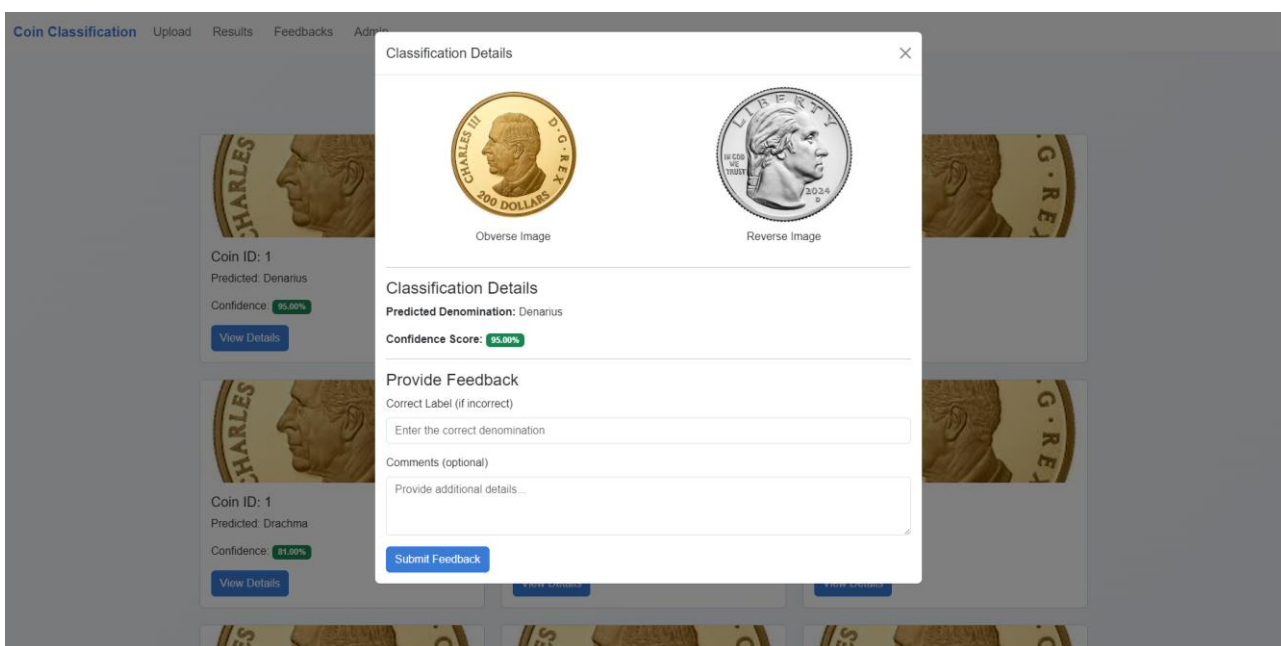
3.3 pav. Monetos įkėlimo puslapis, po įkėlimo gauti rezultatai

Rezultatų puslapis (3.4 pav. ir 3.5 pav.) skirtas peržiūrėti monetų klasifikacijos rezultatus ir pateikti grįžtamąjį ryšį apie klasifikacijos kokybę. Puslapyje rodomos kortelės, kuriose pateikiama pagrindinė informacija apie klasifikacijos rezultatus: monetos ID, numatyta vertė ir pasitikėjimo balas. Kortelės dinamiškai sugeneruojamos iš serverio pateiktų duomenų.

Vartotojas gali paspausti mygtuką „View Details“, kad atidarytų modalinį langą, kuriame pateikiama detalesnė informacija apie monetą, tokia kaip jos nuotrauka, numatyta klasifikacija ir pasitikėjimo balas. Modaliniame lange taip pat suteikiama galimybė vartotojui pateikti grįžtamąjį ryšį, nurodant teisingą etiketę ar pridėdant komentarus. Visi atnaujinimai siunčiami į serverį, o rezultatai atnaujinami realiuoju laiku.



3.4 pav. Rezultatų puslapis, kuriame pateikiama monetų klasifikacija

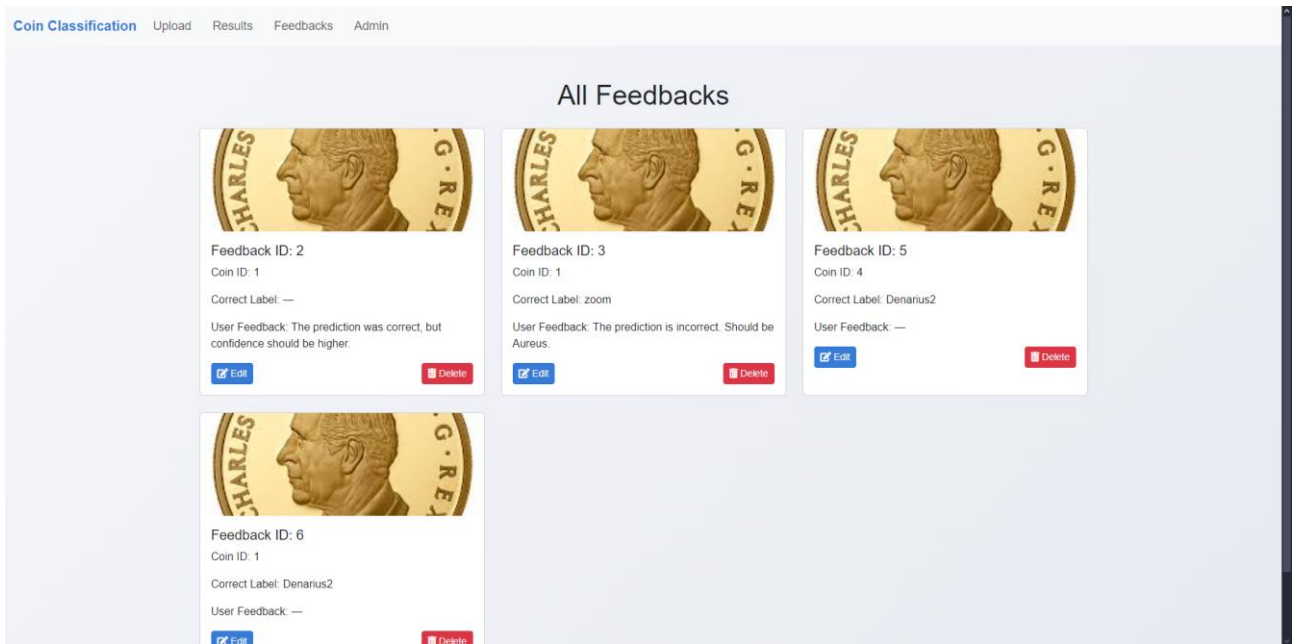


3.5 pav. Rezultatų puslapis, monetos informacija ir atsiliepimo palikimas

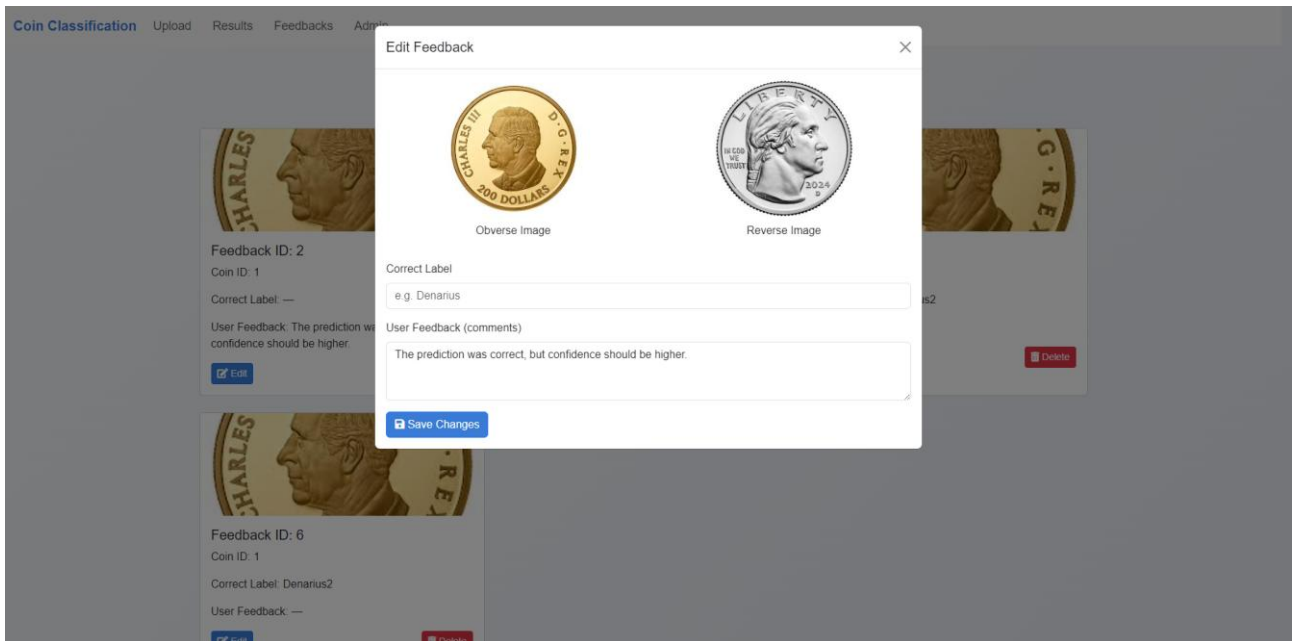
Atsiliepimų puslapis (3.6 pav. ir 3.7 pav.) suteikia vartotojams galimybę peržiūrėti ir redaguoti pateiktus grįžtamąjį ryšį apie monetų klasifikacijos rezultatus. Puslapyje rodomos atsiliepimų kortelės, kuriose pateikiama informacija apie konkretaus atsiliepimo turinį, tokį kaip pateikta teisinga etiketė ar vartotojo komentarai.

Vartotojas gali pasirinkti redaguoti atsiliepimą, o modaliniame lange atnaujinti pateiktą informaciją, pavyzdžiui, pakeisti teisingą etiketę ar pridėti papildomus komentarus. Pakeitimai išsaugomi serverio bazėje ir realiuoju laiku atnaujinami puslapyje.

Be to, vartotojas turi galimybę ištrinti pateiktus atsiliepimus, jei jie nebereikalingi.



3.6 pav. Atsiliepimų peržiūros ir redagavimo puslapis



3.7 pav. Atsiliepimų redagavimas

4. Monetų kandidatų paieškos sprendimo projektavimas ir įgyvendinimas

4.1. Kontekstas ir sprendimo idėja

Darbe sprendžiamas monetų identifikavimo uždavinys. Pagal monetos priekinės ir galinės pusės nuotraukas ir pagal atskirai prie monetos įrašo pateiktą papildomą informaciją, pavyzdžiui, svorį, skersmenį ir metų intervalą, sistema turi pasiūlyti kelis labiausiai tinkančius kandidatus. Klasė čia apibrėžiama kaip monetos denominacija, todėl uždavinys yra 45 klasių atpažinimas rinkinyje, kuriame dalis klasių turi daug pavyzdžių, o dalis labai mažai. Dėl to sprendimas, vertinamas vien bendru tikslumu, dažnai būna patikimas dažnoms klasėms, bet praranda praktinę vertę retoms klasėms, kurios sudaro reikšmingą duomenų dalį.

Dėl to monetų identifikavimas šiame darbe formuluojamas ne tik kaip vieno atsakymo parinkimas, bet ir kaip paieškos pagal panašumą uždavinys. Sistema pateikia kelių kandidatų sąrašą. Jame rodomi keli labiausiai į užklausą panašūs kandidatai, surikiuoti pagal bendrą balą, o ne keli tikimybių įverčiai. Tokia formuluo­­tė yra artimesnė realiam naudojimui. Net ir tada, kai pirmas pasiūlytas variantas yra teisingas, naudotojui dažnai reikia platesnio konteksto. Jei pirmas variantas neteisingas, teisingas atsakymas vis dar gali būti sąrašė.

Sprendimas sudarytas iš trijų dalių.

Pirma, kiekvienas monetos priekinės ir galinės pusės vaizdas paverčiamas požymių vektoriumi, tai yra skaičių sąrašū, pagal kurį galima matuoti vaizdų panašumą. Pagal šiuos vektorius pavyzdžių rinkinyje sudaromas pradinis kandidatų sąrašas. Antra, kandidatų eilė pakeičiama pagal metus, svorį ir skersmenį. Šie laukai padeda atskirti monetas, kurios vaizduose atrodo panašios. Trečia, pagal pirmo ir antro kandidato balų skirtumą vertinama, ar sistema gali pateikti vieną atsakymą, ar geriau palikti kandidatų sąrašą be galutinio pasirinkimo.

Toks padalijimas padeda pamatyti, kurioje vietoje atsiranda klaida. Jei teisinga denominacija nepatenka į kandidatų sąrašą, problema prasideda vaizdo paieškoje. Jei ji yra sąrašė, bet lieka ne pirmoje vietoje, reikia taisyti eilės keitimą pagal papildomus laukus. Jei pirmi kandidatai labai panašūs, reikia tikrinti atsakymo patikimumą. Dėl to vėlesniuose eksperimentuose galima atskirti, ką pakeitė vaizdo paieška, ką pakeitė papildomi monetos laukai ir ką pakeitė atsakymo nepateikimo taisyklė.

Literatūros ir duomenų analizė lėmė keturis sprendimo pasirinkimus. Dėl netolygaus klasių pasiskirstymo vien bendro tikslumo neužtenka, todėl naudojamas makro F1 ir rezultatai atskirai rodomi dažnoms, vidutinėms ir retoms klasėms. Kadangi sprendimas pateikia kandidatų sąrašą, reikia matuoti, ar teisinga moneta patenka tarp kelių pirmų kandidatų. Tam naudojamas atkūrimas@K. Kadangi metai, svoris ir skersmuo gali būti netikslūs arba nepilni, jie tik keičia kandidatų eilę, o ne iš karto atmeta kandidatus. Kadangi sistema kartais gali būti neužtikrinta, atsakymo nepateikimo riba parenkama validacijoje, o testas naudojamas tik galutiniam patikrinimui.

Trumpa šio poskyrio sprendimo suvestinė pateikta 4.1 lentelėje.

4.1 lentelė. Konteksto ir sprendimo idėjos suvestinė

Problema	Projektinis sprendimas	Kaip tai tikrinta
----------	------------------------	-------------------

Daug vizualiai panašių klasių	Vietoje vieno atsakymo naudojama paieška pagal panašumą ir kandidatų sąrašas.	Lyginti bazinis klasifikatorius, prototipai ir kNN paieška, kad matytųsi, kiek duoda vien vaizdo paieška.
Papildoma informacija triukšminga arba nepilna	Papildoma informacija tik pakoreguoja kandidatų vietas sąrašė, bet jų neatmeta.	Atliktos komponentų patikros, pogrupių analizė ir trūkstumų bandymai, kad būtų matyti, kada ši informacija padeda, o kada ne.
Ne visi atvejai vienodai aiškūs	Įdėtas patikimumo sluoksniš ir taisyklė, kada sistema neturėtų pateikti atsakymo.	Vertinta pagal tai, kiek užklausų sistema atsako, atkūrimą@5 ir rizikos santraukas, kad matytųsi, ar sistema geba neparodyti galutinio atsakymo, kai jis per nepatikimas.
Dalis idėjų gali būti brangesnės arba nestabilios	Sudėtingesni sprendimai palikti kaip atskiros patikros, o ne įtraukti į bazinį variantą.	Atlikti kandidato rinkinio dydžio, mokomo perrikuotojo, kelių pavyzdžių ir vaizdo koduotojų bandymai, kad būtų aišku, kurie sprendimai verti papildomo sudėtingumo.

4.2. Reikalavimų specifikacija

Našumo reikalavimas reiškia, kad sistema turi būti pakankamai greita naudoti per naršyklę. Šis reikalavimas laikomas įvykdytu, jei apdorojant bent 100 užklausų iš to paties pavyzdžių rinkinio vidutinis vienos užklausos laikas neviršija 2 s, o 95 procentilio laikas neviršija 5 s.

Kiekvienai užklausiai turi būti galima atkurti visą sprendimo eigą nuo pradinio kandidatų sąrašo iki galutinio atsakymo. Šis reikalavimas laikomas įvykdytu, jei pagal vieną užklausos identifikatorių 100 procentų patikrintų atvejų galima parodyti pradinį kandidatų sąrašą, perrikuotą sąrašą, sprendimą, ar sistema turi pateikti atsakymą, ir naudotojo atsiliepimą.

Rezultatų atkartojimas čia vertinamas pagal tai, ar tie patys rezultatai gaunami naudojant tą patį duomenų variantą, atsitiktinę sėklą ir tuos pačius modelio nustatymus. Šis reikalavimas laikomas įvykdytu, jei dar kartą paleidus tą patį vertinimą su tais pačiais nustatymais gaunamos tos pačios prognozės ir tos pačios galutinės metrikos testavimo aibėje.

Atsparumas nepilniems duomenims reiškia, kad sistema turi veikti ir tada, kai dalies papildomos informacijos nėra. Šis reikalavimas laikomas įvykdytu, jei visoms patikrintoms užklausoms su trūkstamais papildomais laukais vis tiek sugeneruojamas kandidatų sąrašas, o kandidatas neatmetamas vien dėl to, kad trūksta papildomos informacijos.

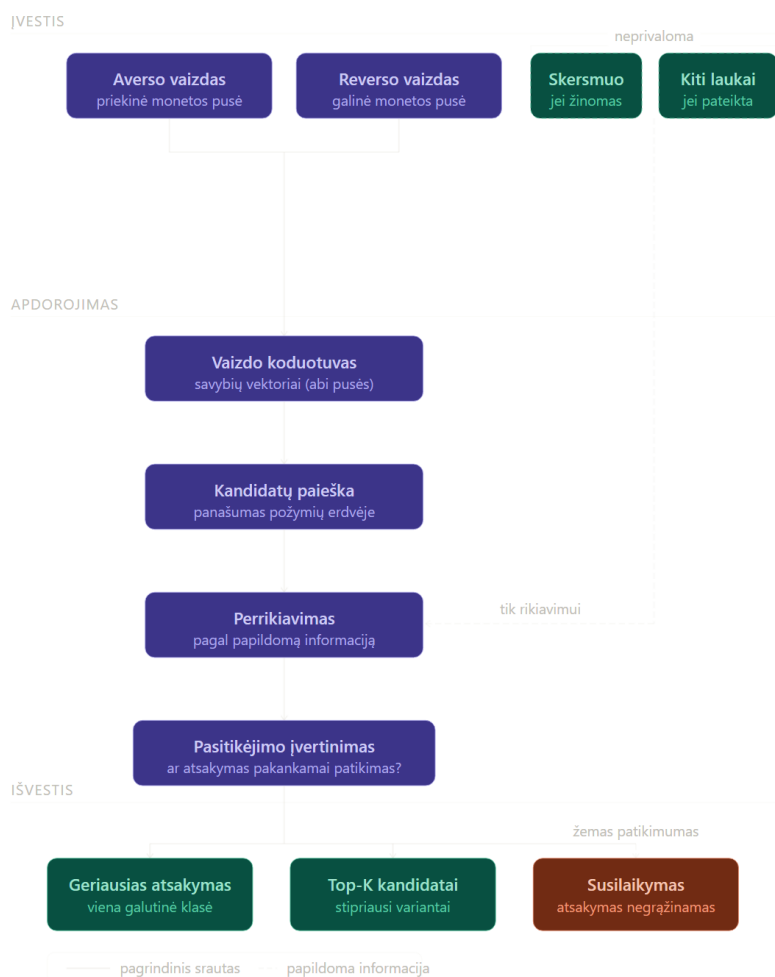
4.3. Bendroji sprendimo architektūra

Sprendimas projektuojamas kaip aiškus duomenų srautas nuo įvesties iki sprendimo.

Bendroji architektūra įgyvendinama šešiais nuosekliais žingsniais. Pirma, paruošiami duomenys. Vaizdai suvienodinami iki to paties dydžio, o papildoma informacija pateikiama pastovia lentele. Trūkstamos reikšmės tvarkomos taip, kad sistema galėtų dirbti ir su nepilnais įrašais. Antra, iš priekinės ir galinės pusės vaizdų išgaunami požymių vektoriai. Trečia, pagal šių vektorių panašumą sudaromas pradinis kandidatų sąrašas ir atrenkamas ribotas kandidatų rinkinys. Ketvirta, kandidatai perrikiuojami derinant vaizdo panašumą su papildomos informacijos sutapimu. Penkta, pagal skirtumą tarp pirmos ir antros vietos balų apskaičiuojama, ar sistema pakankamai tikra. Jei skirtumas per mažas, sistema gali nepateikti atsakymo. Šešta, išsaugomi tarpiniai rezultatai ir ataskaitos, kad vėliau būtų galima tirti klaidas ir palyginti skirtingas sistemos versijas.

Vienos užklausos eiga apibrėžiama atskirai, kad schema būtų tikrinama kaip algoritmas. Iš įvesties vaizdų apskaičiuojamas užklausos požymių vektorius. Tada jo panašumas palyginamas su visais pavyzdžių rinkinio požymių vektoriais. Pagal panašumą atrenkamas pradinis kandidatų rinkinys. Kiekvienam kandidatui apskaičiuojamas bendras balas, kuris jungia vaizdo panašumą ir papildomos informacijos sutapimą. Kandidatai surikiuojami pagal šį balą ir sudaromas galutinis sąrašas. Iš pirmos ir antros vietos balų skirtumo apskaičiuojama, ar sistema pakankamai tikra. Galiausiai išsaugomi kandidatai, balai ir papildomos informacijos įverčiai, kad sprendimą būtų galima paaiškinti ir patikrinti.

4.1 pav. parodyta visa sprendimo eiga nuo įvesties iki galutinio atsakymo. Kiekvienas modulis turi aiškią įvestį, išvestį ir vietą bendrame duomenų sraute, todėl sprendimo kokybę galima matuoti atskirais etapais.



4.1 pav. Sprendimo srauto schema

4.4. Duomenų analizė

Šiame poskyryje paaiškinama, kokie duomenys buvo paruošti sprendimui ir kokias praktines išvadas iš jų gavome. Naudotas MANTIS duomenų rinkinys[30]. Iš RDF aprašų paimta tai, kas gali padėti atpažinti monetą praktikoje. Tai monetos metų intervalo pradžia ir pabaiga, valdovas ar institucija, vertė, monetos leidybos vieta, skersmuo ir svoris. Šie laukai palikti todėl, kad vien vaizdo ne visada pakanka, ypač kai monetos labai panašios.

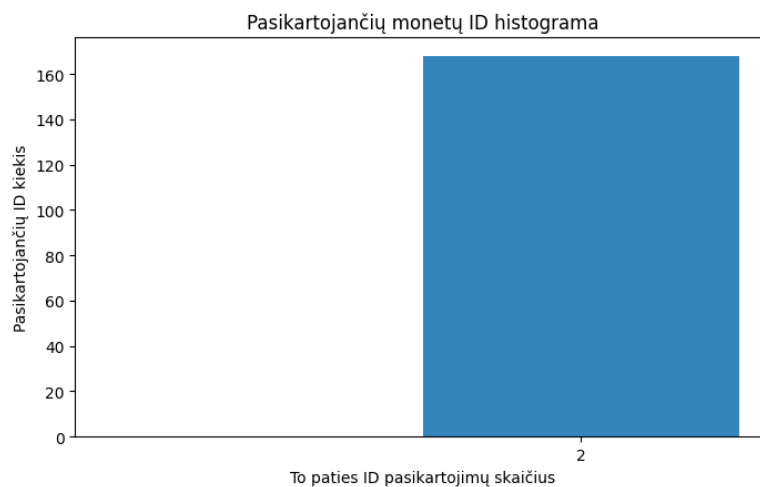
Duomenys iš RDF buvo ištraukti taip, kad galutiniame sprendime liktų du dalykai. Pirmą, patys vaizdai. Antra, keli atskirai prie įrašo pateikti papildomos informacijos laukai, kurie gali padėti atskirti artimus atvejus. Praktikoje tai reiškia, kad sistema remiasi dviem informacijos tipais. Vaizdas yra pagrindinis šaltinis, o papildoma informacija tik padeda patikslinti kandidatų tvarką.

Papildoma informacija čia reiškia ne tai, kas papildomai atpažįstama iš vaizdo, o kartu su monetos įrašu pateiktus laukus. Dalies šių laukų reikšmės gali būti apytikslės, ypač metų ribos, o dalies laukų gali trūkti. Dėl to įgyvendinant sprendimą buvo numatyta, kad papildoma informacija gali padėti tik tada, kai ji yra pateikta ir logiškai suderinama, o kai jos nėra, sistema vis tiek turi veikti, remdamasi vaizdu.

Svarbiausi paruošimo sprendimai susiję su trūkstamomis reikšmėmis, duplikatais ir nevienodu klasių dydžiu (kai dalis klasių turi daug pavyzdžių, o dalis – labai mažai). Atliekant pirminę patikrą nustatyta, kad svorio trūksta labai retai (apie 0,1 %), todėl tai nėra pagrindinis ribojimas. Tuo tarpu skersmens trūksta dažnai (apie 47 %), todėl jo šalinimas reikštų didelę informacijos praradimą ir dar labiau apsunkintų retų klasių atpažinimą.

Dėl šios priežasties trūkstamas skersmuo buvo paliktas kaip nežinoma reikšmė. Projektavimo prasme tai reiškia du dalykus. Pirma, sprendime skersmuo negali būti privalomas kriterijus, nes beveik pusei įrašų jis būtų nepasiekiamas. Antra, vertinant rezultatą būtina atskirai matyti, kaip sistema elgiasi tada, kai skersmuo žinomas ir kai jis nežinomas. Tai vėliau ir buvo daroma pogrupių analizė. Vėlesniuose eksperimentuose ši problema atsispindi kaip aiškus ribojimas. Kai monetos įrašė trūksta dalies papildomos informacijos, sprendimas labiau remiasi vaizdu ir metų informacija.

Dublikatai buvo tvarkomi pagal jų pobūdį. Aptikus pasikartojančius numismatic_object identifikatorius, visiškai sutampantys įrašai (tikslūs duplikatai) buvo pašalinti, nes jie neduoda naujos informacijos ir gali iškraipyti vertinimą. Daliniai pasikartojimai palikti, nes realiuose kataloguose tas pats objektas gali turėti kelis įrašus su skirtingomis detalėmis, pavyzdžiui, skirtingai patikslintu metų intervalu ar kitu papildomos informacijos lauku. Pasikartojančių ID pasiskirstymas parodytas 4.2 pav. Šis sprendimas svarbus ir vertinimui. Duomenų dalijimas į mokymo, parinkimo ir testavimo aibes turi vengti situacijų, kai labai artimi arba tie patys įrašai atsiranda skirtingose aibėse, nes tuomet rezultatas būtų dirbtinai per geras.



4.2 pav. Monetų identifikatorių pasikartojimų histograma duomenų rinkinyje

4.2 pav. parodyta, kad dauguma identifikatorių pasikartoja tik kelis kartus, o labai dažnai pasikartojančių įrašų yra nedaug. Tai svarbu, nes pasikartojantys įrašai gali dirbtinai pagerinti rezultatą, jei labai panašūs objektai patenka į skirtingas aibes. Po duplikatų patikros toliau vertinami skaitiniai laukai, todėl 4.3 pav. pateikiamas monetų svorio pasiskirstymas.



4.3 pav. Monetų svorio pasiskirstymo histograma

4.3 pav. parodyta, kad dauguma monetų yra lengvos, o labai sunkių monetų yra nedaug. Tai reiškia, kad svoris gali padėti atskirti dalį atvejų, bet vien jo nepakanka visoms klasėms atskirti. 4.4 pav. tuo pačiu principu pateikiamas skersmens pasiskirstymas.

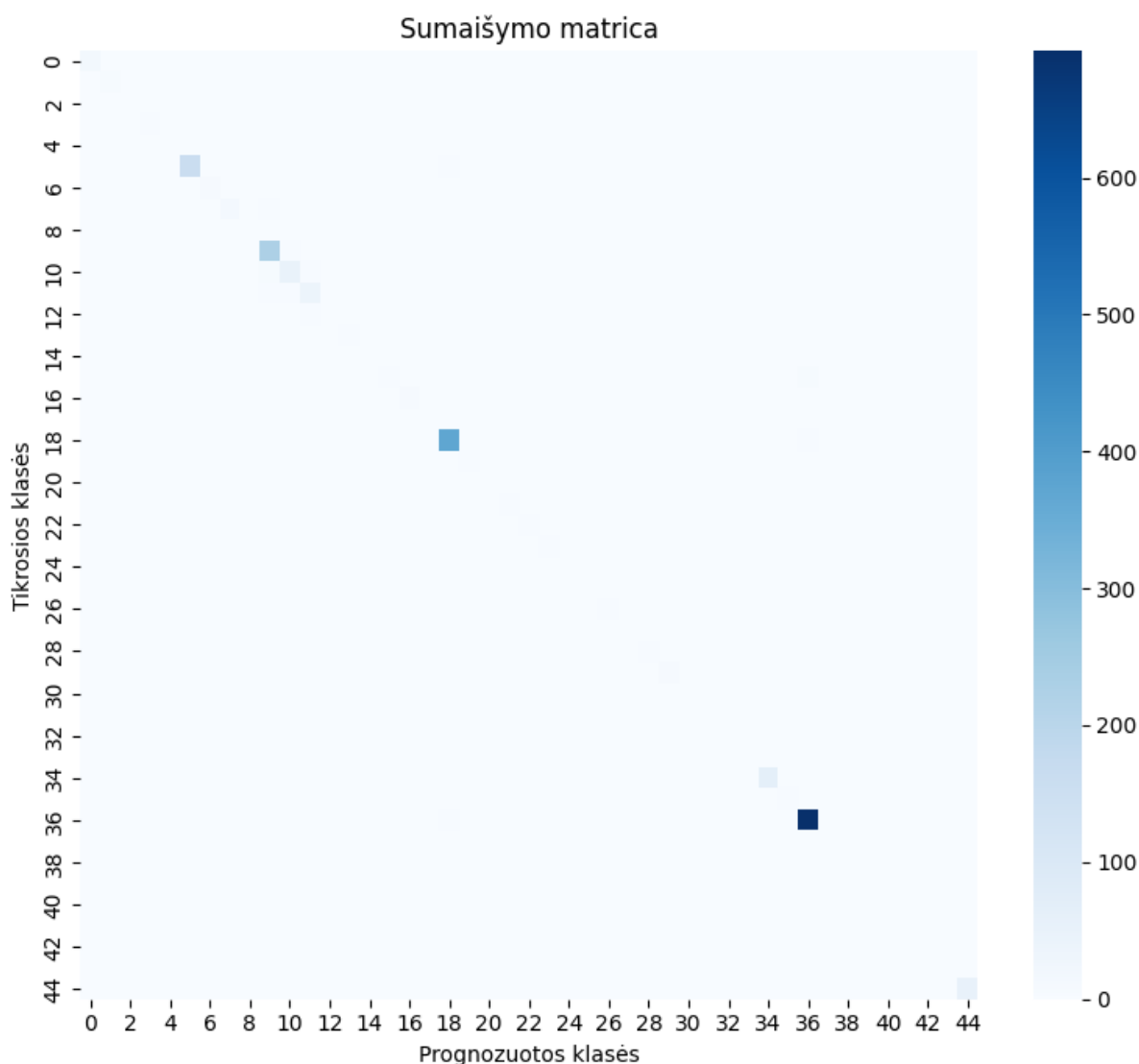


4.4 pav. Monetų skersmenų pasiskirstymo histograma

4.4 pav. parodyta, kad skersmenys telkiasi gana siaurame intervale, todėl šis laukas gali padėti, bet vien jo nepakanka artimoms klasėms atskirti. Ši išvada dar svarbesnė todėl, kad beveik pusei įrašų skersmens trūksta, o klasių aprašai dažnai persidengia. Dėl to papildoma informacija šiame darbe naudojama ne kaip griežtas atmetimas, o tik kandidatų tvarkai pakoreguoti.

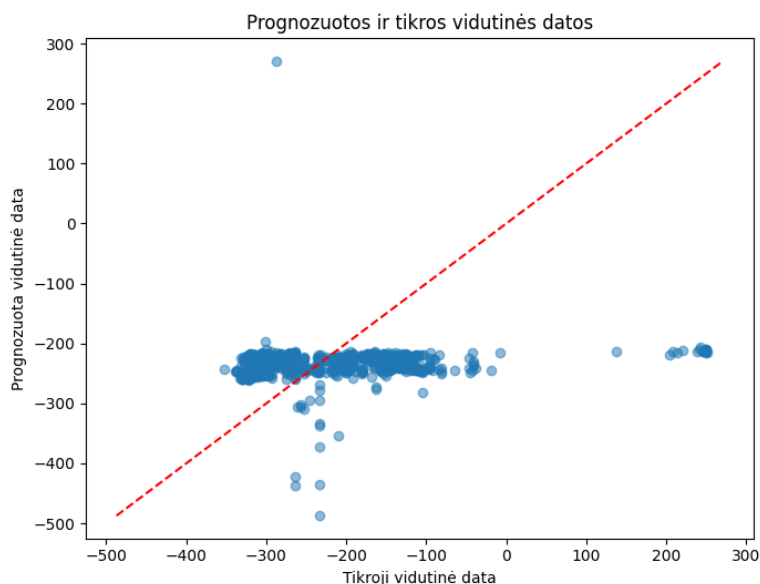
Kategorinių laukų, tai yra valdovo, vertės ir monetos leidybos vietos, pasiskirstymas taip pat yra nevienodas. Nedidelė dalis kategorijų pasitaiko labai dažnai, o didelė dalis retai. Tai svarbu todėl, kad net jei modelis vidutiniškai dirba gerai, jis gali klysti būtent ten, kur duomenų mažiausia.

Buvo atlikta pradinė patikra su paprastu modeliavimu, naudojant vien skaitinius laukus, tai yra svorį, skersmenį, pradžios metus ir pabaigos metus. Logistinės regresijos bandyme, prognozuojant monetos vertę, gautas apie 0,71 tikslumas mokyme ir apie 0,70 tikslumas teste. Tačiau klaidų analizė parodė tipinę netolygaus rinkinio problemą. Dažnesnės klasės klasifikuojamos gerokai geriau, o retos dažnai painiojamos (4.5 pav.).



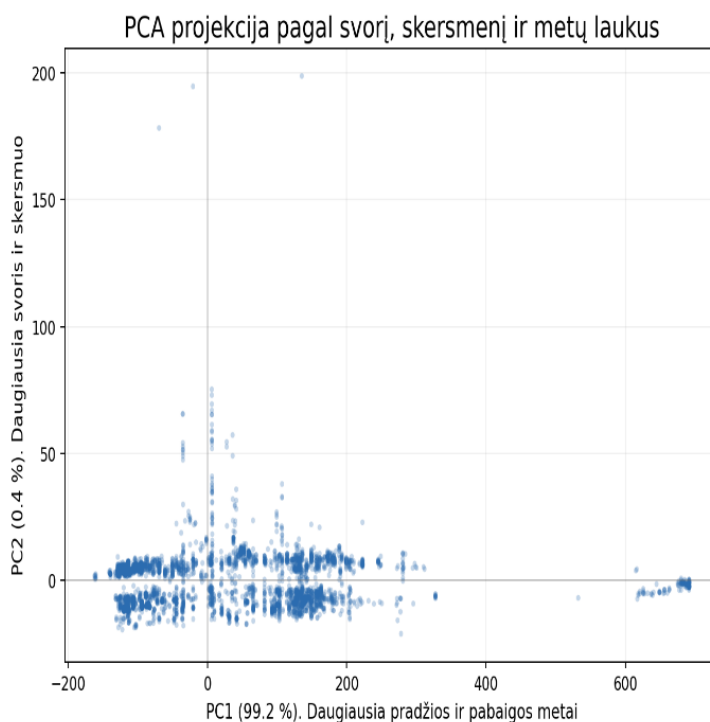
4.5 pav. Klaidų sumaišymo matrica

Ši matrica rodo, kurias klases modelis, naudodamas vien svorį, skersmenį ir metų laukus, supainioja dažniausiai. Ji taip pat rodo, kad vien šių laukų neužtenka patikimai atskirti panašias klases. Dėl to šiame darbe jie naudojami tik kaip papildoma informacija po vaizdo paieškos. 4.6 pav. papildomai parodo, kad iš šių laukų apskaičiuota data su tikrąja sutampa tik apytikriai.



4.6 pav. Sklaidos diagrama, kurioje lyginama iš svorio, skersmens ir metų laukų apskaičiuota monetos vidutinė data ir tikroji vidutinė data

Iš svorio, skersmens ir metų laukų apskaičiuota vidutinė data su tikrąja sutampa silpnai. Jei šių laukų pakaktų monetos laikotarpiui nustatyti, taškai būtų susitelkę arti įstrižainės, bet jie išsibarstę daug plačiau. Todėl 4.7 pav. ir 4.8 pav. toliau rodo, kaip tie patys skaitiniai laukai atrodo sumažinti iki dviejų ašių.



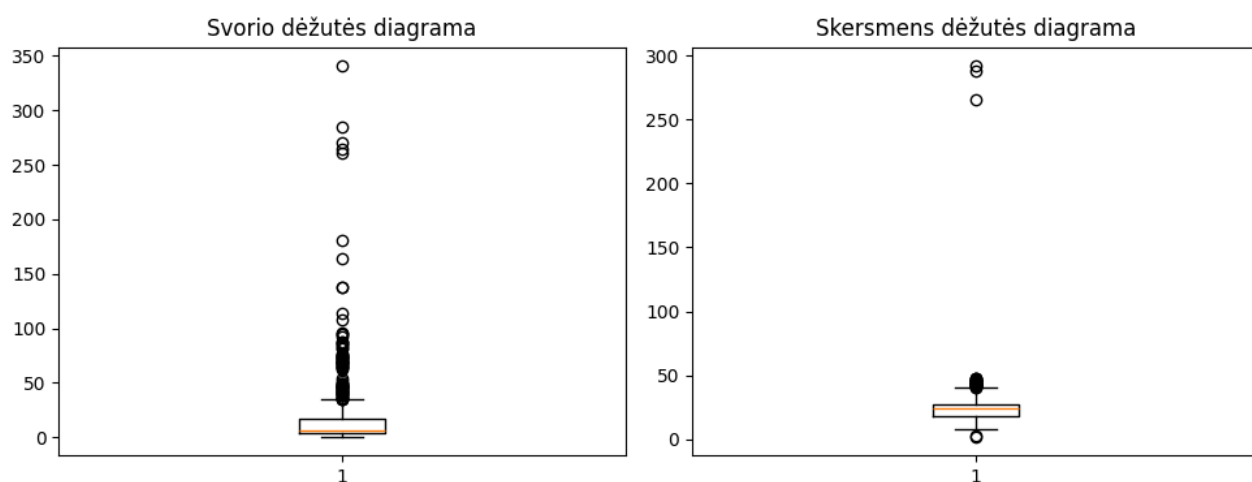
4.7 pav. Sklaidos diagrama, kurioje monetų skaitiniai laukai, taikant pagrindinių komponentų analizę, sumažinti iki dviejų ašių

4.7 pav. Sklaidos diagrama, kurioje monetų skaitiniai laukai, taikant pagrindinių komponentų analizę, sumažinti iki dviejų ašių parodė, kaip tie patys keturi skaičiais pateikti požymiai išdėstomi dviejose

pagrindinėse ašyse. Pirmą ašį daugiausia lemia pradžios ir pabaigos metai, ir ji paaiškina apie 99,2 % šių laukų skirtumų. Antrą ašį labiau lemia svoris ir skersmuo, tačiau ji paaiškina tik apie 0,4 %. Iš to aiškėja, kad datos laukai turi didžiausią įtaką, o vien iš svorio, skersmens ir metų laukų aiškių klasių grupių nesusidaro.

4.8 pav. Sklaidos diagrama, kurioje tie patys monetų skaitiniai laukai t-SNE būdu išdėstyti dvimatėje erdvėje

4.8 pav. tie patys skaitiniai laukai parodyti t-SNE būdu, tai yra metodu, kuris panašius įrašus padeda išdėstyti arčiau vienus kitų dvimatėje erdvėje. Tačiau iš šio vaizdo negalima spręsti, kiek skirtumų paaiškina kiekviena ašis. Aiškiai atskirtų klasių nesusidaro, todėl skaitiniai laukai labiau tinka kaip papildomas signalas, o ne kaip pagrindinis atskyrimo šaltinis. 4.9 pav. šią analizę papildė svorio ir skersmens dėžučių diagramomis.



4.9 pav. Monetų svorio ir skersmens dėžučių diagramos

Apibendrinant, duomenų analizė tiesiogiai pagrindžia pasirinktą sistemos logiką. Pirma sprendimas remiasi vaizdu ir suformuoja kelių siūlomų variantų sąrašą. Tada reitingas patikslinamas papildoma informacija, tai yra metais, skersmeniu ir svoriu, jeigu ji yra pateikta. Tokia seka yra praktiška ir todėl, kad ji nesulaužo sprendimo, kai dalies papildomos informacijos trūksta. Kai papildoma informacija nepilna, sistema vis tiek gali pasiūlyti kelis vizualiai artimus variantus, o vėlesniame etape papildoma informacija veikia kaip pagalba, o ne kaip būtina sąlyga.

Šis poskyris taip pat apibrėžia aiškius ribojimus, kurie vėliau atsispindi eksperimentinėje dalyje ir išvadose. Pirma, daliai įrašų trūksta skersmens, todėl papildoma informacija ne visada gali padėti. Antra, kategorinių laukų pasiskirstymas yra nevienodas, todėl sistemą reikia vertinti taip, kad būtų matoma, kaip jai sekasi ir su retomis klasėmis. Trečia, skaitiniai laukai vieni nėra pakankami, todėl pagrindinis sprendimo pagrindas yra vaizdinė paieška, o papildoma informacija tik patikslina reitingą.

Trumpa duomenų analizės suvestinė pateikta 4.2 lentelėje.

4.2 lentelė. Duomenų analizės suvestinė

Ką parodė duomenys	Kodėl tai svarbu	Koks sprendimas dėl to priimtas
4 706 iš 10 008 įrašų (47,0 %) trūksta skersmens.	Todėl skersmens negalima naudoti kaip griežto filtro, nes taip būtų atmesta daug dar tinkamų kandidatų.	Skersmuo naudojamas tik kaip papildomas signalas, kai jis yra pateiktas.
Svorio trūksta tik 10 iš 10 008 įrašų (0,1 %).	Todėl svoris gali padėti beveik visose užklausose, bet vien jo nepakanka atskirti panašias monetas.	Svoris naudojamas perrikiuojant kandidatus, bet ne kaip vienintelis sprendimo pagrindas.
Mokymo aibėje mažiausia klasė turi 3, mediana 12, o didžiausia 2 101 pavyzdį.	Toks skirtumas reiškia, kad vien bendras tikslumas gali paslėpti silpnus retų klasių rezultatus.	Rezultatai rodomi ir pagal dažnumo grupes, ne tik vienu bendru skaičiumi.
PCA parodė, kad pirma ašis paaiškina 99,2 %, o antra tik 0,4 % svorio, skersmens ir metų laukų skirtumų.	Tai rodo, kad metų laukai turi didžiausią įtaką, bet vien iš svorio, skersmens ir metų laukų aiškių klasių grupių nesusidaro.	Pagrindinis sprendimas remiamas vaizdo paieška, o skaitiniai laukai naudojami tik po jos.

4.5. Duomenų paruošimas ir pateikimas sistemai

Darbe naudoti duomenys buvo paimti iš Nomisma duomenų rinkinių puslapio [30], kuriame pateikiamos su numizmatika susijusios susietųjų duomenų kolekcijos, tarp jų ir MANTIS įrašai. Iš šių RDF įrašų buvo paimti laukai, reikalingi praktinei monetų atpažinimo sistemai. Tai priekinės ir galinės pusės vaizdų nuorodos, denominacijos etiketė ir pasirinkti papildomos informacijos laukai, tokie kaip metų intervalas, svoris, skersmuo, o kai kuriuose variantuose ir valdovas ar monetas leidybos vieta. Buvo pašalinti įrašai su trūkstama priekine arba galine puse, atliktas deduplikavimas pagal objekto identifikatorių ir pagal identišką vaizdų porą, pašalinti konfliktiniai duplikatai ir

paliktos tik klasės, turinčios bent 5 pavyzdžius. Iki valymo rasta 336 įrašai su pasikartojančiu identifikatoriumi ir 382 pasikartojančios priekinės ir galinės pusės poros, todėl deduplikavimas čia buvo būtinas.

Pagrindiniame duomenų variante rinkinys suskirstytas į tris dalis. Tai mokymo, validavimo ir testavimo aibės. Kiekviena aibė paruošta kaip fiksuotos struktūros masyvai, atskirai saugant priekinės ir galinės pusės vaizdus bei papildomos informacijos laukus. Šiame variante yra 5 634 mokymo pavyzdžiai, 1 882 validavimo pavyzdžiai ir 1 882 testavimo pavyzdžiai. Klasių, tai yra denominacijų, skaičius yra 45. Padalijimas yra fiksuotas ir naudojamas visuose palyginimuose, kad rezultatų skirtumai būtų siejami su pakeistu sprendimu, o ne su atsitiktiniu duomenų perskirstymu.

Norint suprasti, ar duomenys tinkami šiam darbui, reikia aiškiai įvardyti trūkstamą papildomą informaciją ir matavimo vienetus. Šiame rinkinyje skersmuo yra dažniausiai trūkstamas laukas, jo nėra apie 47 % įrašų. Svorio trūksta retai, apie 0,1 % įrašų. Priekinės arba galinės pusės vaizdų nuorodos trūksta apie 2 % įrašų. Tokie įrašai pašalinami prieš sudarant aibes. Skaitiniai laukai naudojami paprastais vienetais. Svoris matuojamas gramais, skersmuo milimetrais, o monetos laikotarpis metais, tai yra intervalu nuo pradžios iki pabaigos metų.

Papildomi monetos laukai šiame darbe yra kelių rūšių. Svoris ir skersmuo pateikiami skaičiais. Laikotarpis pateikiamas pradžios ir pabaigos metais. Kiti laukai, pavyzdžiui, valdovas ar autoritetas, yra tekstiniai. Svoris, skersmuo ir metai suvienodinami pagal skalę, kad vienas didesniais skaičiais matuojamas laukas neužgožtų kitų. Kai reikšmės trūksta, įrašoma neutrali reikšmė ir kartu pridedama žymė, kad laukas nežinomas. Pagrindiniame duomenų variante naudojami keturi suvienodinti skaičių laukai ir viena žymė, ar skersmuo žinomas.

Šiame variante skaičiais pateiktų požymių vektorius sudarytas iš penkių dalių. Į jį įeina normalizuotas svoris, normalizuotas skersmuo, normalizuota pradžios metų reikšmė, normalizuota pabaigos metų reikšmė ir žymė, ar skersmuo trūksta.

Papildomai numatyti keli maži vaizdų paruošimo bandymai. Jais tikrinama, ar sprendimas nesugriūva nuo smulkių vaizdo paruošimo pakeitimų. Pagrindinis sprendimas remiasi vaizdo panašumo paieška ir metų, svorio bei skersmens naudojimu po paieškos. Todėl vaizdų paruošimo pakeitimai laikomi antriniais. Jei jie padeda, juos galima pridėti kaip papildomą žingsnį, bet jie neturi pakeisti pagrindinės paieškos ir perrikiavimo sekos.

Praktikoje šie bandomieji variantai parodė, kad agresyvus pradinis vaizdų apdorojimas gali būti žalingas. Kontrasto pagerinimas ir triukšmo mažinimas stabilaus pagerėjimo nedavė, o fono šalinimas kartais pablogino paiešką, nes po tokio apdorojimo panašios monetos nebebuvo taip patikimai randamos tarp artimiausių kandidatų. Todėl šiame darbe pagrindinis pagerėjimas siejamas su paieškos ir papildomos informacijos perrikiavimo logika, o pradinis vaizdų apdorojimas paliekamas tik kaip antrinė ir rizikingesnė tolesnio tobulinimo kryptis.

Kad duomenų patikimumas būtų valdomas praktiškai, vaizdų paruošimas turi visada vykti ta pačia tvarka. Ta pati tvarka taikoma mokymo, validavimo ir testavimo aibėms. Darbe laikomasi trijų pagrindinių principų.

Pirma, visi vaizdai suvienodinami iki fiksuoto dydžio (224×224), o jų skaičiai suvienodinami pagal tą pačią skalę. Antra, vaizdo paruošimo žingsniai yra iš anksto nustatyti. Jei kuris nors žingsnis

nepavyksta, klaida įrašoma, o ne praleidžiama be žymės. Trečia, apdoroti vaizdai ir papildomi požymiai išsaugomi kaip pastovūs paketų masyvai. Dėl to vėlesni eksperimentai remiasi ta pačia įvestimi, o rezultatų skirtumai siejami su keičiamu sprendimu, ne su atsitiktinai iš naujo paruoštais duomenimis.

Mokymo metu taikomas duomenų didinimas (augmentacija), tačiau jis netaikomas validavimo ir testavimo aibėms, kad vertinimas atspindėtų realius duomenis. Lengvame duomenų didinimo režime naudojamos atsitiktinės rotacijos iki maždaug 20° , poslinkiai iki maždaug 10 %, mastelio keitimas iki maždaug 10 % ir horizontalus atspindys. Kraštai užpildomi artimiausia reikšme. Stipresniame režime papildomai įtraukiami šlyties ir ryškumo pokyčiai. Duomenų didinimo režimas laikomas keičiamu nustatymu, todėl jo įtaka gali būti tikrinama komponentų patikromis, o galutiniai nustatymai užfiksuojami prieš pateikiant testavimo rezultatus.

Svarbi šio uždavinio savybė yra didelis netolygumas net ir po duomenų sutvarkymo, tai yra po deduplikavimo ir minimalaus klasės pavyzdžių skaičiaus reikalavimo. Bendroje aibėje klasių dydžiai svyruoja nuo 5 iki 3 497 pavyzdžių, o mokymo aibėje klasių dydžiai gali būti labai nevienodi, nuo kelių pavyzdžių iki daugiau nei 2 000. Tai reiškia, kad vertinant sistemą būtina naudoti metrikas, kurių neužgožia dažniausios klasės.

Kad didelio netolygumo poveikis būtų matomas ne tik bendrai, projektavimo dalyje numatyta analizuoti rezultatus ir pagal klasių grupes, sudarytas pagal mokymo pavyzdžių skaičių. Taip galima įvertinti ne tik tai, ar sistema pagerėjo vidutiniškai, bet ir ar ji pagerėjo ten, kur užduotis sunkiausia. Praktikoje klasės grupuojamos į tris grupes.

Šiame darbe klasės skirstomos į tris grupes pagal pavyzdžių skaičių mokymo aibėje. Dažnos klasės turi bent 100 pavyzdžių. Vidutinės klasės turi nuo 20 iki 99 pavyzdžių. Retos klasės turi nuo 1 iki 19 pavyzdžių.

Toks skaidymas svarbus aiškinant rezultatus. Pavyzdžiui, makro F1 pagerėjimas be retų klasių grupės pagerėjimo praktiškai reikštų, kad sistema geriau atpažįsta dažnas denominacijas, bet nepagerina sprendimo ten, kur užduotis sunkiausia.

Validavimo ir testavimo aibės šiame darbe naudojamos griežtai atskiriant modelio parinkimus nuo galutinio įvertinimo. Validacijoje parenkami nustatymai, pavyzdžiui, kandidatų rinkinio dydis, perrikiavimo svoriai ir patikimumo slenkstis. Testavimo aibėje vieną kartą pateikiami jau užfiksuoti rezultatai.

Kad būtų aišku, iš kur atsiranda rezultatų skirtumai, naudoti keli duomenų variantai. Bazinis variantas buvo skirtas patikrinti, ar visa seka veikia ir ar rodikliai skaičiuojami teisingai. Variantas su deduplikavimu ir minimalaus klasės pavyzdžių skaičiaus reikalavimu sumažino riziką, kad tie patys vaizdai pateks į kelias aibes, ir padėjo gauti lengviau aiškinamus rezultatus pagal klases. Variante su skaliavimu be duomenų nutekėjimo skalės suvienodinimo skaičiai buvo imami tik iš mokymo aibės. Taip validavimo ir testavimo duomenys nebuvo netyčia priderinti prie mokymo. Taip pat buvo sudaryti atskiri variantai papildomiems klausimams, pavyzdžiui, su trūkstamumo žymėmis ar su monetos leidybos vieta.

Šie duomenų variantai buvo reikalingi tam, kad rezultatų skirtumus būtų galima sieti su konkrečiu pakeitimu, o ne su atsitiktiniu duomenų paruošimu. Bazinis variantas parodė, ar visa seka apskritai

veikia. Variantas su deduplikavimu ir minimalaus klasės pavyzdžių skaičiaus reikalavimu sumažino kraštutinių atvejų įtaką. Variantas su skaliavimu be duomenų nutekėjimo leido tiksliau įvertinti papildomą informaciją. Papildomi variantai su trūkstumų žymėmis ar monetos leidybos vieta buvo naudojami tik atskiriems klausimams patikrinti.

Tokie variantai leidžia projektavimo dalyje aiškiai parodyti, kad rezultatų skirtumai kyla iš konkretaus sprendimo pakeitimo, pavyzdžiui, iš papildomos informacijos apdorojimo, o ne iš atsitiktinių duomenų paruošimo pokyčių.

Galiausiai, svarbu įvardyti, ką reiškia viena užklausa šiame darbe. Viena užklausa susideda iš dviejų vaizdų, tai yra monetos priekio ir galo, ir papildomos informacijos įrašo. Tai svarbu, nes monetos priekis ir galas dažnai turi skirtingą informatyvumą. Vienoje pusėje gali būti valdovo atvaizdas, o kitoje gali būti vertę nurodantis tekstas ir kiti identifikaciniai ženklai. Dėl to dviejų pusių naudojimas yra esminė projektinė prielaida.

Trumpa duomenų paruošimo ir jų formos suvestinė pateikta 4.3 lentelėje.

4.3 lentelė. Duomenų paruošimo ir jų formos suvestinė

Paruošimo žingsnis	Kodėl reikėjo	Nuo ko tai apsaugo
Pašalinti tikslūs duplikatai.	Tokie įrašai neduoda naujos informacijos ir gali dirbtinai pagerinti rezultatą.	Sistema negauna klaidingai per lengvo vertinimo.
Atskirta, kad labai artimi įrašai nepatektų ir į mokymą, ir į testą.	Jei tas pats objektas pasirodytų abiejose vietose, rezultatas atrodytų geresnis negu yra iš tikro.	Taip išvengiama duomenų nuotėkio tarp aibių.
Trūkstamas skersmuo paliktas kaip nežinoma reikšmė, o ne išmestas.	Skersmens trūksta per dažnai, todėl jo šalinimas reikštų didelį duomenų praradimą.	Išvengiama bereikalingo duomenų sumažinimo, ypač retose klasėse.
Tie patys paruošimo žingsniai taikyti visoms aibėms.	Tik taip galima sąžiningai lyginti validacijos ir testavimo rezultatus.	Rezultatų skirtumai labiau atspindi modelį, o ne skirtingą paruošimą.

4.6. Požymių vektorių išgavimas

Vaizdo požymiams išgauti šiame darbe pasirinktas MobileNetV2 [31]. Tai lengvas konvoliucinis tinklas, sukurtas taip, kad vaizdo požymius būtų galima išgauti greitai ir be didelės skaičiavimo

kainos. Ši architektūra skaičiavimus išskaido į paprastesnes operacijas. Dėl to modelis išlaiko svarbią vaizdo informaciją, bet lieka mažesnis už didesnes architektūras. Šiam darbui tai reiškia, kad MobileNetV2 leidžia gana greitai gauti naudingus monetų požymius ir nereikalauja daug skaičiavimo išteklių. Tai buvo svarbu todėl, kad čia požymių vektorius reikia gauti ne iš vieno, o iš dviejų monetos vaizdų, o po to naudoti juos paieškoje su visu pavyzdžių rinkiniu. Kitaip tariant, modelis turėjo būti ne tik pakankamai tikslus, bet ir pakankamai lengvas, kad visas sprendimas nesulėtėtų vien dėl vaizdo dalies.

Į palyginimą taip pat įtrauktas MobileNetV3Small [32]. Tai naujesnis ir dar mažesnis tos pačios krypties modelis. Jis sukurtas tam, kad iš labai nedidelio modelio būtų išgaunama kuo daugiau naudos. Dėl to kai kurios jo dalys perstatytos taip, kad modelis geriau išnaudotų savo ribotą dydį. Šiame darbe MobileNetV3Small buvo reikalingas ne kaip pagrindinis kandidatas, o kaip patikra, kiek dar galima mažinti modelį neprarandant per daug kokybės. Norėta pamatyti, ar dar mažesnis modelis vis dar išlaiko pakankamai informatyvius monetų požymius, ar taupymas jau pradeda per daug kenkti rezultatui. Tai buvo svarbu todėl, kad monetų vaizduose daug smulkių detalių. Tokios detalės kaip valdovo atvaizdo kontūrai, raidžių dalys ar nedideli ženklai gali lemti, ar teisinga moneta pateks į kandidatų sąrašą. Jei net toks mažas modelis būtų pasirodęs pakankamai gerai, tai būtų stiprus argumentas rinktis dar lengvesnį sprendimą. Jei ne, tada aiškiau matyti, kur baigiasi saugus modelio mažinimas.

Stipresnei palyginimo kryptčiai buvo pasirinkti EfficientNetB0 [33] ir EfficientNetV2B0 [34]. EfficientNet šeimos pagrindinė mintis yra ne didinti tik vieną modelio dalį. Kartu derinama, kiek sluoksnių turi modelis, kiek informacijos jis apdoroja vienu metu ir kokio dydžio vaizdą gauna. Dėl to modelis paprastai auga tvarkingiau negu tada, kai didinamas tik vienas iš šių dalykų. EfficientNetV2 tęsia tą pačią kryptį, bet dalį sprendimų pakeičia taip, kad modelį būtų patogiau ir greičiau mokyti. Šiame darbe abu variantai buvo naudingi todėl, kad leido patikrinti ne vieną atsitiktinį stipresnį modelį, o visą modelių šeimą, kuri literatūroje dažnai laikoma geru tikslumo ir skaičiavimo kainos kompromisu. Monetų uždavinyje toks palyginimas svarbus todėl, kad reikia atpažinti labai smulkius vaizdo skirtumus. Dėl to buvo pagrįsta patikrinti, ar stipresni bendros paskirties modeliai duoda aiškų laimėjimą prieš lengvesnius mobiliuosius variantus. Jei jų pranašumas mažas, tada lengvesnis modelis lieka praktiškesnis pasirinkimas. Jei pranašumas didelis, tada jau verta svarstyti, ar didesnė kaina nėra pateisinama.

Į architektūrų palyginimą dar įtrauktas ResNet50V2 [35]. Tai gilesnis modelis, kuriame dalis informacijos gali keliauti trumpesniu keliu, neapkraudama kiekvieno tarpinio žingsnio. Toks sprendimas padeda stabiliau mokyti gilesnį tinklą. V2 versijoje kai kurios modelio dalys perrikiuotos taip, kad mokymas būtų dar stabilesnis. ResNet50V2 atstovauja kryptčiai, kuri remiasi ne modelio mažumu, o gebėjimu naudoti daugiau sluoksnių neprarandant stabilumo. Šiame darbe jis buvo reikalingas kaip aiškus palyginimo taškas už lengvųjų mobiliųjų modelių ribų. Taip tikrinta, ar didesnis ir gilesnis modelis monetų paieškoje duoda tiek naudos, kad verta mokėti didesnę skaičiavimo kainą. Toks klausimas svarbus todėl, kad vien šiek tiek geresnis rezultatas lentelėje dar nereiškia geresnio galutinio sprendimo. Jei modelis tampa per lėtas ar per sunkus, visas paieškos kelias gali tapti mažiau praktiškas.

Visa ši architektūrų grupė buvo pasirinkta ne atsitiktinai. Ja siekta palyginti tris aiškias krypttis. Pirmoji krypttis yra lengvas pagrindinis mobilus modelis. Antroji yra dar mažesnis tos pačios kryptties

variantas. Trečioji yra stipresni bendros paskirties modeliai iš kitų šeimų. Toks rinkinys leidžia matyti ne tik tai, kuris modelis surinko daugiausia konkrečioje lentelėje. Jis leidžia matyti ir tai, kaip rezultatai keičiasi didėjant modelio dydžiui, sudėtingumui ir skaičiavimo kainai. Šiame darbe toks palyginimas buvo svarbus dėl dviejų priežasčių. Pirma, reikėjo pasirinkti modelį, kuris būtų pakankamai stiprus paieškai, bet kartu neišpūstų viso sprendimo vykdymo laiko. Antra, reikėjo parodyti, kad galutinis pasirinkimas neatsirado vien todėl, kad buvo įprasta naudoti vieną modelių šeimą. Jis turėjo remtis realiu palyginimu tarp skirtingų architektūrų. Dėl to MobileNetV2 paliktas kaip pagrindinis variantas, o kitos architektūros naudotos kaip kryptingas kontrolinis palyginimas. Toks palyginimas padeda įvertinti, kiek papildomo rezultato duoda didesni ar kitokie vaizdo modeliai ir ar tas laimėjimas šiame darbe iš tikrųjų vertas papildomos kainos.

Požymių vektoriai šiame darbe gaunami iš prieš klasifikaciją esančio požymių sluoksnio. Iš kiekvienos vaizdo pusės išgaunamas globalaus vidurkio telkimo vektorius (angl. Global Average Pooling), o priekinės ir galinės pusės vektoriai sujungiami į vieną 2 560 ilgio požymių vektorių. Šis vektorius naudojamas paieškai, nes jis yra fiksuoto ilgio ir leidžia greitai skaičiuoti panašumus su visu pavyzdžių rinkiniu.

Eksperimentinėje dalyje parodyta, kad vienos monetos pusės vaizdo informacijos dažnai pakanka K kandidatų sąrašui sudaryti, tačiau jos per mažai stabiliai išskirti teisingą denominaciją į pirmąją poziciją netolygiame scenarijuje (žr. Eksperimentą 3a). Todėl projektavimo išvada yra tiesioginė. Dviejų vaizdų, tai yra priekinės ir galinės pusės, naudojimas nėra šalutinis priedas. Tai esminė prielaida, kurią patvirtina komponentų patikra.

Tolesniuose darbuose būtų galima keisti, kiek svarbos gauna kiekviena monetos pusė. Jei vienos pusės vaizdas yra nekokybiškas, sistema galėtų daugiau remtis kita puse, o ne visada jungti abi puses vienodai.

Šiame darbe požymių vektorių erdvė yra pagrindinis vaizdo informacijos šaltinis, bet ji viena sprendimo nenulemia. Pirmiausia pagal ją sudaromas kandidatų sąrašas. Po to papildoma informacija tik pakeičia kandidatų tvarką. Toks skaidymas turi tris praktinius pranašumus.

Pirma, toks skaidymas atsparus papildomos informacijos trūkumui. Jei dalis papildomos informacijos nepateikiama, paieškos modulis vis tiek veikia. Antra, jį lengviau paaiškinti. Jei atsiranda klaida, galima atskirti, ar problema yra paieškos erdvėje, ar perrikiavimo taisyklėje. Trečia, jį lengviau valdyti didėjant mastui. Paiešką galima optimizuoti atskirai, pavyzdžiui, keičiant kandidatų rinkinio dydį, nepriklausomai nuo papildomos informacijos logikos.

Šis etapas remiasi prielaida, kad požymių vektorių erdvėje vizualiai panašios monetos atsiduria arti viena kitos. Tada paieška pagal panašumą gali sudaryti prasmingą kandidatų sąrašą. Ribojimas yra tas, kad vien vaizdo dažnai nepakanka smulkiems skirtumams, pavyzdžiui, monetos laikotarpiui ar kitiems papildomos informacijos skirtumams, atskirti. Todėl paieška pagal vaizdo panašumą (naudojant požymių vektorius) šiame darbe laikoma pirmu etapu, o ne galutiniu sprendimu.

Papildomas ribojimas yra duomenų pobūdis. Monetų vaizdai gali skirtis apšvietimu, fono triukšmu, pasukimu ir kokybe. Tai svarbu todėl, kad dėl tokių skirtumų požymių vektorių erdvėje atsiranda vizualiai panašių, bet neteisingų kaimynų. Dėl to nukenčia ir kandidatų sąrašo atranka, ir klaidų tipas, pavyzdžiui, teisingos klasės nebuvimas kandidatų sąrašė.

4.7. Kandidatų paieška pagal požymių vektorius

Paieškos etapas naudoja požymių vektorių panašumą tarp užklauskos pavyzdžio ir pavyzdžių rinkinyje esančių pavyzdžių. Panašumo skaičiavimo būdas parenkamas taip, kad tiktų požymių vektoriams. Praktikoje naudojamas kosinusinis panašumas. Jis patogus, nes gerai dera su suvienodintais vektoriais ir leidžia tiesiogiai palyginti, kiek du požymių vektoriai panašūs.

Paieškos rezultatas yra reitinguotas kandidatų sąrašas, tačiau projektavimo sprendime aiškiai atskiriami du reitingai.

Pirma, vizualus reitingas pagal panašumą požymių vektorių erdvėje. Jis naudojamas kandidatų atrankai į rinkinį. Antra, galutinis reitingas po papildomos informacijos perrikiavimo. Jis naudojamas ir galutiniam atsakymui pagal pirmąją poziciją.

Toks atskyrimas padeda aiškiau tirti klaidas. Jei teisingas kandidatas nepatenka į sąrašą, perrikiavimas jau nepadės. Jei kandidatas sąrašo yra, bet nukrenta žemyn, problema greičiausiai yra perrikiavimo signaluose.

Projektavimo dalyje buvo svarbu apibrėžti, kaip iš paieškos gaunamas galutinis atsakymas pagal pirmąją poziciją, kad būtų galima jį palyginti su klasifikacija. Čia nagrinėtos kelios sprendimo taisyklės.

Pirma, prototipų klasifikatorius. Kiekvienai klasei skaičiuojamas požymių vektorių vidurkis, o prognozuojama klasė parenkama pagal didžiausią panašumą šiam vidurkiui. Privalumas yra greitis ir paprastumas, o trūkumas tas, kad vienas vidurkis blogai atspindi klases, kurios turi didelę vizualinę įvairovę. Antra, kNN balsavimas. Čia klasė parenkama pagal artimiausią kaimyną arba pagal kelių kaimynų balsavimą. Siekiant sumažinti vieno nugalėtojo efektą, naudojamas balsavimas su svoriais, kai kaimynų įtaka nustatoma pagal panašumo funkciją.

Balsavimo su svoriais idėja projektuojama taip, kad būtų aiški ir atkartojama. Kaimynų svoriai skaičiuojami pagal panašumą, o ne parenkami ranka kiekvienam atvejui atskirai.

$$w_i = \exp(\text{sim}_i / t) / \sum_j \exp(\text{sim}_j / t),$$

kur sim_i yra kosinusinis panašumas, o t yra jautrumo nustatymas. Jis valdo, ar balsavimas bus artimas griežtam variantui, kai t mažas, ar labiau išsklaidytas, kai t didesnis. Tokia formuluotė patogiai, nes ji susieja sprendimo taisyklę su aiškiu nustatymu, kurį galima parinkti validacijoje.

Papildomai buvo tikrinta klasikinė paieškos taisyklė, vadinama užklauskos išplėtimu (angl. query expansion, pseudo-relevance feedback). Čia užklauskos požymių vektorius pakeičiamas taip, kad į jį patektų ir artimiausių kaimynų vidurkis. Apskritai vaizdų paieškoje tai kartais padeda, bet šiame monetų uždavinyje po tokio žingsnio reitingo kokybė suprastėjo. Tai rodo, kad toks vidurkinimas gali sumaišyti labai artimų klasių ribą.

Papildomai buvo patikrinta kontekstinės paieškos idėja, artima įžvalgoms apie požymių vektorių mokymą pagal panašumą ir kontekstinį panašumą. Šiam bandymui naudotas 20 artimiausių kaimynų kontekstas. Čia kandidatai rikiuojami ne vien pagal tiesioginį kosinusinio panašumo įvertį. Kandidato balas dar koreguojamas pagal tai, kiek sutampa užklauskos ir kandidato artimiausių kaimynų rinkiniai. Mintis tokia. Jei užklausa ir kandidatas turi panašų vietinį kontekstą požymių vektorių erdvėje, jie

turėtų būti labiau suderinti. Praktikoje šiame darbe tai įgyvendinta kaip lengvas perrikiavimas kandidatų rinkinio viduje, kai kandidato balas šiek tiek pakoreguojamas pagal bendrą kaimynų dalį.

Šiame smulkaus detalumo monetų scenarijuje rezultatas buvo neigiamas. Nors kNN ir papildomos informacijos perrikiavimo logika liko ta pati, vieno atsakymo kokybė po perrikiavimo suprastėjo. Tai rodo, kad kaimynų sutapimo idėja čia veikia panašiai kaip užklauskos išplėtimas. Ji sustiprina bendrą panašumą, bet susilpnina skirtumą tarp labai artimų denominacijų.

Papildomai buvo tikrinta taisyklė, kai k reikšmė keičiama iš 1 į 3 tada, kai sprendimas nėra aiškus. Paieškos scenarijuje dažna situacija, kai pirmos ir antros vietos kandidatai turi labai panašius balus. Tada griežtas vieno kandidato sprendimas tampa nestabilus. Todėl įtraukta taisyklė, kuri k reikšmę keičia pagal situaciją. Jei skirtumas tarp pirmos ir antros vietos balų mažas, sprendimas sudaromas iš kelių kaimynų, o ne vieno. Taip sumažinamas vieno pavyzdžio atsitiktinumo poveikis.

K reikšmė parodo kompromisą tarp naudotojo darbo ir sistemos naudingumo. Mažas K , pavyzdžiui 1, imituoja klasifikaciją, o didesnis K , pavyzdžiui 5 ar 10, atitinka paieškos sistemos scenarijų, kai naudotojas peržiūri kelis variantus. Eksperimentinėje dalyje K naudojamas ir kaip rekomendacijų rodiklio atkūrimas@ K pagrindas, ir kaip praktinės sistemos išvesties apibrėžimas.

Kandidatų sąrašo dydis yra sprendimo dalis, o ne vien bandymo nustatymas. Jis tiesiogiai lemia, kiek kandidatų bus vertinama su papildoma informacija, ir kartu nustato, kiek perrikiavimas apskritai gali padėti. Todėl kandidatų sąrašo dydis parenkamas kaip kompromisas tarp kokybės ir skaičiavimo kainos.

Čia svarbu ir skaičiavimo kaina. Jei panašumas skaičiuojamas su visu pavyzdžių rinkiniu, vienos užklauskos kaina didėja kartu su pavyzdžių rinkiniu. Todėl didėjant duomenų kiekiui reikia aiškiai suprasti, kur atsiranda pagrindinės sąnaudos. Jų yra trys. Tai požymių vektorių išgavimas, panašumo skaičiavimas su visu pavyzdžių rinkiniu ir papildomas kandidatų rinkinio perrikiavimas. Papildomos informacijos perrikiavimas taikomas tik atrinktiems kandidatams, todėl jo kaina yra kontroliuojama. Toks skaidymas leidžia realistiškai planuoti sistemos mastelį. Jei ateityje pavyzdžių rinkinys didės, bus galima optimizuoti paieškos dalį nekeičiant papildomos informacijos logikos ir vertinimo tvarkos.

Papildomai mastelio valdymas gali būti sprendžiamas mažinant patį pavyzdžių rinkinį, tai yra taikant kandidatų atranką arba retinimą (angl. pruning). Tai rizikingiau nei keisti kandidatų sąrašo dydį. Jei retinimas parinktas blogai, jis gali pašalinti pavyzdžius, kurie reikalingi klasės vidinei įvairovei parodyti, ypač retoms klasėms. Bandomieji variantai parodė, kad paprastas retinimas pagal klasės centrą stipriai blogina rezultatą. Jis palieka tik arti centro esančius pavyzdžius ir pašalina tos pačios klasės įvairovę, kuri smulkių monetų skirtumų paieškoje yra labai svarbi. Atsitiktinė atranka kenkė mažiau, bet jos rezultatas priklausė nuo atsitiktinės sėklos, todėl vien toks retinimas nėra pakankamai stabilus sprendimas.

Tuo tarpu netolygų pasiskirstymą labiau atitinka retinimas, kai ribojamas tik dažnų klasių perteklius. Tokiu atveju paliekami visi retų klasių pavyzdžiai, o tik dažnos klasės apkarpos iki viršutinės ribos. Šis principas leidžia reikšmingai sumažinti pavyzdžių rinkinį neprarandant didžiosios dalies kokybės. Todėl jis pagrįstas kaip ateities kryptis, jei pavyzdžių rinkinys vėliau didėtų.

Netolygumo kontekste atsiranda dar viena praktinė problema. Net jei pavyzdžių rinkinys nėra labai didelis, dažnų klasių pavyzdžiai gali per dažnai patekti į kandidatų sąrašo viršų vien dėl savo gausos. Tai reiškia, kad kandidatų sąrašė gali būti per mažai skirtingų klasių, o tai ypač kenkia vidutinėms ir retoms klasėms. Todėl buvo patikrinta taisyklė, kuri didina kandidatų sąrašo įvairovę. Pirmą apskaičiuojamas didelis kNN kandidatų sąrašas, o tada jis sutrumpinamas iki galutinio kandidatų skaičiaus, ribojant, kiek kandidatų galima paimti iš vienos klasės. Ši taisyklė nekeičia paieškos rodiklio ir kandidatų neatmeta griežtai, bet mažina per didelę dažnų klasių įtaką sąrašo viršuje.

Projektavimo metu taip pat buvo apsvarstytos mažos rizikos paieškos taisyklės, kurios nenaudoja papildomo mokymo, pavyzdžiui, užklausos išplėtimas ir klasių dažnio korekcijos kNN balsavime. Visgi jos nebuvo įtrauktos į pagrindinę sprendimo seką. Šios idėjos įvertintos atskirai eksperimentinėje dalyje, kad sistema išliktų lengvai paaiškinama. Pirmiausia veikia vaizdo paieška, tada eina perrikiavimas, o galiausiai vertinama, ar sistema pakankamai tikra.

Paieškos etapas čia turi atlikti vieną pagrindinį darbą, tai yra įtraukti teisingą klasę į kandidatų sąrašą. Jei to neįvyksta, vėlesni žingsniai jau nebegali pataisyti rezultato.

4.8. Papildomos informacijos perrikiavimas

Papildomos informacijos perrikiavimo etapas prideda kandidatams papildomą balą, kuris rodo, kiek kandidato papildoma informacija dera su užklausos papildoma informacija. Šiame darbe naudoti du pagrindiniai papildomos informacijos signalai.

Pirma, svorio ir skersmens atitikimas. Svorio ir skersmens artumas naudojamas kaip nedidelis papildomas įvertis. Kuo mažesnis skirtumas, tuo didesnis papildomas balas. Kai skersmens reikšmė trūksta, tai nelaikoma griežtu neatitikimu. Vietoje to papildomos informacijos įtaka sumažinama per trūkstumų žymę. Antra, metų intervalo atitikimas. Kai papildomoje informacijoje pateikiamas metų intervalas, kandidato ir užklausos intervalai lyginami pagal persidengimą ir bendrą sutapimą. Papildomas balas pritaikomas taip, kad pagal vaizdą panašūs, bet pagal laikotarpį netinkami kandidatai neužimtų sąrašo viršaus.

Perrikiavimo balas sudaromas kaip svorinis vaizdinio ir papildomos informacijos signalų derinys.

Praktiškai tai reiškia, kad papildoma informacija tik pakoreguoja kandidatų vietas sąrašė, bet kandidatų neatmeta. Vaizdo panašumo balas užtikrina, kad kandidatai būtų vizualiai artimi, o papildoma informacija padeda tada, kai keli kandidatai atrodo panašiai. Šis etapas yra perrikiavimas, o ne filtravimas, kad sistema išliktų atsparesnė netiksliesiems ar nepilniems papildomos informacijos įrašams.

$$S(c) = w_{img} \cdot sim_{img}(c) + w_{num} \cdot s_{num}(c) + w_{date} \cdot s_{date}(c).$$

Papildomai prieinami kategoriniai laukai, pavyzdžiui, valdovas ar monetos leidybos vieta, gali būti įtraukti kaip atskiros atitikimo dalys. Pagrindiniuose eksperimentuose jų svoriai buvo lygūs 0, todėl galutiniame balo skaičiavime jie nenaudoti.

Papildomai patikrintas Reciprocal Rank Fusion, arba RRF. Šiame variante kandidatai jungiami pagal jų vietas keliuose sąrašuose, o ne pagal pačių balų dydį. Priedo patikroje toks sujungimas davė tik

0,6479 testavimo makro F1, kai bazinis balų sumavimas siekė 0,8190. Todėl sprendime paliekamas balų sumavimas su validacijoje parinktais svoriais.

Kadangi papildomos informacijos gali trūkti ir realiame naudojime, projektavimo dalyje numatyta atsparumo patikra su dirbtinai padidintu trūkstumumu validacijoje. Ši patikra vėliau atlikta Eksperimente 3e. Ji leidžia patikrinti, ar nustatymai netampa per trapūs tada, kai papildomos informacijos yra mažiau.

Papildomai atliktas bandomasis patikrinimas su kategorine papildoma informacija, tai yra monetos leidybos vieta. Jis parodė, kad toks laukas gali duoti tik ribotą naudą kaip nedidelis papildomas įvertis. Griežtas atmetimas, kai reikalaujama sutapimo ir atmetami likę kandidatai, pasirodė rizikingas. Triukšmingas arba nevienodai užpildytas papildomos informacijos laukas gali pašalinti teisingą kandidatą taip, kad vėlesni žingsniai jo nebesugrąžins. Todėl kategorinė papildoma informacija laikoma tik papildoma kryptimi. Jei ji naudojama, tai tik kaip mažo svorio perrikiavimo dalis, o ne kaip atmetimo taisyklė.

Papildomai patikrintos trys perrikiavimo korekcijos iš literatūros. Pirma korekcija naudojo klasės ir papildomos informacijos ryšį. Ji tikrino, ar konkrečiai klasei būdingas svoris, skersmuo ar metų intervalas turėtų padidinti kandidato balą. Antra korekcija naudojo vieno atsakymo klasifikatoriaus softmax balą kaip pradinį kandidato įvertį. Trečia korekcija trūkstamas papildomos informacijos reikšmes suvedė į bendrą nežinomą reikšmę, kad jos būtų traktuojamos vienodai. Nė viena kryptis stabilaus pagerėjimo nedavė, o softmax pradinio įverčio priedai rezultatą aiškiai blogino. Todėl šios korekcijos neįtrauktos į pagrindinį sprendimą. Papildoma informacija toliau jungiama per aiškius sutapimo balus su validacijoje parinktais svoriais.

Čia $sim_img(c)$ yra vaizdo panašumas, o $s_num(c)$ ir $s_date(c)$ yra papildomos informacijos sutapimo balai. Praktinė šios formulės nauda yra trijų rūšių. Pirma, galima aiškiai paaiškinti, kodėl kandidatas pakilo ar nusileido. Antra, galima atlikti komponentų patikras išjungiant atskiras dalis. Trečia, galima valdyti kompromisą tarp vaizdo signalo ir papildomos informacijos signalo. Trūkstamoms reikšmėms taikoma logika, kuri atsižvelgia į trūkumą. Jei konkretus laukas trūksta, atitinkama formulės dalis nepadaro kandidato automatiškai blogu. Ji tik sumažina tai, kiek papildoma informacija tuo atveju gali padėti atskirti kandidatus.

Kad perrikiavimo dalys būtų ne tik idėja, bet ir aiškiai aprašomas algoritmas, papildomos informacijos sutapimas skaičiuojamas iš kelių paprastų požymių. Svorio ir skersmens laukams pirmiausia suvienodinama skalė, apskaičiuota mokymo aibėje ir pritaikyta visoms aibėms. Tuomet skaičiuojamas atstumas tarp užklauso ir kandidato reikšmių, pavyzdžiui, absoliučių skirtumų suma. Šis atstumas paverčiamas balu mažėjančia funkcija. Kuo atstumas mažesnis, tuo $s_num(c)$ didesnis. Toks aprašymas pasirinktas todėl, kad jį lengva patikrinti. Jis leidžia valdyti papildomos informacijos įtaką svoriu w_num ir nereikalauja mokyti papildomo modelio vien tam, kad svorio ir skersmens skirtumai būtų įtraukti į sąrašą.

Metų intervalo signalui skaičiuojamas atskiras atitikimo balas $s_date(c)$. Jei užklauso ir kandidato intervalai gerai persidengia, kandidatas gauna didesnę įvertį. Jei intervalai aiškiai nesutampa, kandidato įvertis mažinamas. Šiame darbe naudotas intervalų persidengimo santykis, panašus į Jaccard rodiklį. Jei metų laukas trūksta arba yra netvarkingas, šis balas negali tapti griežta bausme, nes tada teisingas kandidatas būtų nustumtas vien dėl nepilnų duomenų.

Perrikiavimo funkcija svarbi todėl, kad ji leidžia paiešką valdyti. Vietoje to, kad sąrašo tvarką visiškai lemtų vaizdo panašumas, įvedamas papildomas balas iš domeno informacijos, tai yra svorio, skersmens ir metų. Kad šis papildomas balas nebūtų per stiprus, perrikiavimas daromas kaip balų derinys, o svoriai parenkami validacijos aibėje.

Perrikiavimo logiką sudaro dvi paprastos idėjos.

Pirma, papildomos informacijos sutapimas skaičiuojamas kaip balas. Svorio ir skersmens atitikimas nėra tik taip arba ne. Vietoje to naudojamas atstumas, kuris sumažina kandidato balą, jei jo papildoma informacija labai skiriasi nuo užklauso, bet kandidato visiškai nepašalina. To reikia, nes realiuose duomenyse matavimai turi paklaidą, o papildomos informacijos įrašai gali būti netikslūs. Antra, metų informacija naudojama kaip papildoma apsauga. Ji yra viena stipriausių užuominų, tačiau dažnai pateikiama intervalais ir gali būti nepilna. Todėl metų sutapimas veikia atsargiai. Jei kandidato ir užklauso intervalai nesutampa, kandidato balas mažinamas. Kai intervalų kraštai beveik sutampa, griežtas atmetimas netaikomas.

Šiame etape pasirinkta vengti griežto atmetimo, tai yra filtravimo. Nors griežti filtrai atrodo paprasti, pavyzdžiui, galima reikalauti monetos leidybos vietos sutapimo arba datų persidengimo, jie turi dvi rizikas. Pirma, triukšmingas arba nepilnas laukas gali pašalinti teisingą kandidatą. Antra, filtrai daro sistemą trapią, kai tas pats faktas skirtinguose šaltiniuose užrašomas nevienodai. Todėl papildomi monetos laukai naudojami kaip nedidelis balo priedas, o ne kaip taisyklė, kuri kandidatą iš karto pašalina.

Papildomai buvo svarstytas ir mokomas modelis, kuris po paieškos pakeičia kandidatų eilę. Jis turėtų išmokti, kaip kartu naudoti vaizdo panašumą, metus, svorį ir skersmenį. Tokia kryptis teoriškai gali rasti sudėtingesnes taisykles nei rankinė formulė, bet ji didina riziką prisitaikyti prie triukšmo. Tai patikrinta bandomuoju variantu su į retas klases labiau orientuota užklauso atranka, tačiau rezultatas buvo prastesnis nei perrikiavimas pagal paprastas taisykles. Todėl pagrindiniuose sistemos nustatymuose paliktas taisyklinis perrikiavimas, kurio svoriai parenkami validacijoje.

Kad mokomas perrikiavimas būtų stipresnis, papildomai įgyvendintas mokymas pagal visą kandidatų sąrašą, tai yra rikiavimo mokymas. Čia mokymo pavyzdžiai formuojami tik iš realaus K geriausių kandidatų rinkinio ir grupuojami pagal užklauso identifikatorių (`query_id`). Tai leidžia mokytis ne atskiras poras, o pačią kandidatų eilę vienos užklauso kontekste.

Bandomasis rezultatas su LightGBM LambdaRank [22] parodė, kad mokymas pagal visą kandidatų sąrašą šiame etape dar nepranoksta pagal paprastas taisykles paremto papildomos informacijos perrikiavimo. Tai svarbus neigiamas rezultatas. Jis rodo, kad be daugiau ir tikslesnės informacijos apie kandidatus mokomas modelis kandidatų tvarkai keisti nėra savaime geresnis už paprastą, aiškiai paaiškinamą taisyklę.

Papildomi monetos laukai šiame darbe nepanaikina kandidatų iš sąrašo. Jie tik pakeičia jau rastų kandidatų eilę. Taip teisingas kandidatas neprarandamas vien dėl trūkstamo skersmens, netikslaus laikotarpio ar nevienodai užrašytos papildomos informacijos.

4.9. Patikimumas ir kada sistema neturėtų pateikti atsakymo

Patikimumas šiame darbe reiškia sistemos gebėjimą atpažinti atvejus, kai jos sprendimas nėra pakankamai užtikrintas. Pagrindinis patikimumo skaičius yra skirtumas tarp pirmos ir antros vietos balų po perikiavimo. Jei geriausias kandidatas tik nežymiai lenkia antrą, sprendimas laikomas mažiau tikru.

Atsakymo nepateikimo taisyklė valdoma slenksčiu. Jei skirtumas tarp pirmos ir antros vietos balų mažesnis už slenkstį, sistema nepateikia vieno galutinio atsakymo. Tada skaičiuojama, kokiai užklausų daliai sistema atsakė ir kokia buvo likusių atsakymų kokybė. Slenkstis parenkamas validacijos aibėje pagal du reikalavimus. Sistema turi atsakyti bent iš anksto nustatyta užklausų daliai ir tarp atsakytų užklausų turi išlaikyti kuo geresnį atkūrimą@5. Taip galima parodyti, kiek klaidų rizikos sumažėja, jei sunkiausiems atvejams sistema atsakymo nepateikia.

Patikimumo patikra reikalinga ir tada, kai naujos klasės pridedamos po truputį, o naujai klasei turime tik kelis pavyzdžius. Todėl kelių pavyzdžių atidėjimo eksperimente atskirai tikrinta, kada sistemai geriau nepateikti vieno atsakymo. Ten parodyta, kad rekomendacijų kokybę galima padidinti, jei sistema privalo atsakyti tik į iš anksto nustatytą užklausų dalį.

Kad balų skirtumą būtų lengviau aiškinti, papildomai patikrinta kalibruota versija. Skirtumas tarp pirmos ir antros vietos balų paverčiamas tikimybe, kad atsakymas teisingas. Tam naudota Platt kalibracija, tai yra logistinė regresija. Kalibracijos funkcija apmokoma tik validacijos aibėje ir tada be papildomo derinimo taikoma testavimo aibe. Sprendimas atsakymo nepateikti valdomas tikimybės slenksčiu. Tikimybių kokybė vertinama trimis būdais. Brier balas rodo, kiek prognozuota tikimybė skiriasi nuo realaus atsakymo teisingumo. ECE rodo vidutinę tikimybių paklaidą, o MCE rodo didžiausią tikimybių paklaidą.

Atsakymo nepateikimas neturi tapti sistemingu retų klasių atmetimu[29]. Todėl projektuojant numatyta tikrinti ne tik bendrą atsakytų užklausų dalį, bet ir tai, kaip ji pasiskirsto tarp dažnų, vidutinių ir retų klasių grupių. Taip galima pamatyti, ar slenkstis per dažnai nepalieka be atsakymo būtent retų klasių užklausų.

Patikimumo patikra reikalinga dėl dviejų priežasčių. Pirma, K geriausių kandidatų sąrašas pats nepasako, ar pirmas kandidatas pakankamai atsiskyrė nuo antro. Kartais teisingas kandidatas gali būti sąrašė, bet žemai, o viršuje gali būti keli labai panašūs neteisingi variantai. Antra, netolygiame rinkinyje retoms klasėms dažnai trūksta pavyzdžių, todėl sistema dėl jų gali būti mažiau užtikrinta. Tokiais atvejais ji turi mokėti parodyti, kad vienas galutinis atsakymas dar nėra patikimas.

Skirtumas tarp pirmos ir antros vietos balų pasirinktas todėl, kad jį lengva paaiškinti. Jis tiesiogiai atsako į klausimą, ar geriausias kandidatas pakankamai atsiskiria nuo antro geriausio. Tai nėra vienintelis galimas būdas. Galima būtų tikrinti, ar balai sąrašė labai išsisklaidę, ar pirmo ir antro kandidato balų santykis pakankamai didelis, arba ar rezultatas išlieka panašus keičiant kandidatų sąrašo dydį.

Slenksčio parinkimas turi būti atkartojamas. Slenkstis parenkamas validacijos aibėje. Parenkant slenkstį taikoma riba, kokiai mažiausiai užklausų daliai sistema turi pateikti atsakymą. Tik tada tas pats slenkstis perkeliamas į testavimo aibę.

Pirma, validacijos aibėje apskaičiuojamas skirtumas tarp 1 ir 2 vietos balų ir atkūrimas@5 keliems slenksčiams. Antra, parenkamas slenkstis, kuris duoda geriausią atkūrimą@5, bet kartu palieka bent nustatytą užklausų dalį su pateiktu atsakymu. Trečia, tas pats slenkstis be pakeitimų pritaikomas testavimo aibei ir pateikiamas galutinis rezultatas.

Ši procedūra saugo nuo testo priderinimo. Slenkstis pasirenkamas validacijoje, o testavimo aibėje tik patikrinamas. Rezultatas parodo, kokia rekomendacijų kokybė lieka tada, kai sistema atsako bent nustatytai užklausų daliai.

Papildomai numatoma palyginti du atsakymo patikimumo būdus. Vienas naudoja skirtumą tarp pirmos ir antros vietos balų, kitas naudoja kalibruotą tikimybę. Vertinimas atliekamas per rizikos kreivę ir atsakytų užklausų dalį. Šiame darbe rizika reiškia 1 – atkūrimas@5. Atsakytų užklausų dalis reiškia, kokiai visų užklausų daliai sistema nusprendžia pateikti atsakymą. Kreivė parodo, ar patikimesniais laikomi atvejai iš tikrųjų turi mažiau klaidų. AURC yra plotas po šia kreive. Mažesnis AURC reiškia, kad lengvesnės ir sunkesnės užklauskos atskiriamos geriau.

Patikimumo patikra reikalinga tada, kai keli kandidatai atrodo per daug panašūs. Jos vertė matuojama ne vien likusių atsakymų kokybe, bet ir tuo, kiek užklausų sistema palieka be vieno galutinio atsakymo.

4.10. Projektavimo ir įgyvendinimo išvados

Projektavimo dalyje suformuluota sprendimo struktūra apibrėžia kelis pagrindinius pasirinkimus. Monetos identifikavimas čia traktuojamas kaip paieška, o ne vien vieno atsakymo klasifikacija. Metai, svoris ir skersmuo naudojami kandidatų eilei pakoreguoti, o ne griežtam atmetimui. Patikimumas vertinamas pagal atsakytų užklausų dalį ir atkūrimą@5, o slenkstis parenkamas validacijoje ir tik tada taikomas testui.

Toks projektavimas sukuria tyrimo logiką eksperimentinei daliai. Galima atskirai patikrinti, ką duoda vien paieška pagal vaizdo panašumą, kiek prideda metų, svorio ir skersmens naudojimas po paieškos, kaip keičiasi rezultatai valdant kandidatų sąrašo dydį ir ar atsakymo nepateikimo taisyklė sumažina klaidų riziką išlaikant pakankamą atsakytų užklausų dalį. Likę ribojimai nėra slepiami. Nepilni papildomi laukai, trūkstamos reikšmės ir labai netolygios klasės tampa atskiromis eksperimento dalimis.

Šis skyrius parodo eksperimentų logiką. Eksperimentai čia nėra atsitiktinių bandymų rinkinys. Kiekvienas jų tikrina vieną projektinę prielaidą arba vieną konkrečią riziką.

Bazinis klasifikatorius parodo, ką duoda šis požiūris netolygiame scenarijuje. Prototipai ir kNN parodo, ką duoda vien vizualinė paieška ir kur ji stringa. Papildomos informacijos komponentų patikros parodo, kurie struktūriniai signalai yra būtini ir kurie yra antriniai. Kandidatų rinkinio dydžio patikra parodo kompromisą tarp kokybės ir kainos. Patikra, kada sistema neturėtų pateikti atsakymo, parodo, ar galima objektyviai valdyti patikimumą. Mokomo modelio kandidatų tvarkai keisti prototipas leidžia patikrinti, ar sudėtingesnis sprendimas duoda realią naudą, ar tik įveda nestabilumą. Etykečių smulkumo patikra leidžia atskirti modelio klaidas nuo klasių sistemos ar žymėjimo artefaktų. Klaidų analizė susieja rezultatus su duomenų ribojimais ir tolesnio darbo kryptimis.

Projektavimo dalyje taip pat atskirai numatyta grėsmių patikimumui grupė, kuri susijusi ne su modeliu, o su duomenimis. Dalis klasių yra labai artimos, o etikečių smulkumas gali būti per smulkus praktiniam scenarijui. Dėl to projektuojamas atskiras eksperimentinis patikrinimas su stambesnėmis grupėmis be pertreniravimo. Jis leidžia atskirti, kurios klaidos kyla dėl modelio, o kurios atsiranda dėl klasių sistemos arba žymėjimo artefaktų. 4.4 lentelėje apibendrinta, kaip šie projektavimo pasirinkimai susieja duomenų sandarą, vertinimo apibrėžimą ir pasirinktą veiksmų seką.

4.4 lentelė. Projektavimo dalies suvestinė

Sprendimo dalis	Koks sprendimas pasirinktas	Kodėl pasirinktas	Kaip tai patikrinta
Paieška	Pirma sudaromas kandidatų sąrašas pagal vaizdo panašumą.	Vieno atsakymo klasifikacija viena pati per silpna, ypač retoms klasėms.	Lyginti bazinis klasifikatorius, prototipai ir kNN paieška.
Papildoma informacija	Papildomi laukai tik perstato kandidatų eilę, bet jų neatmeta.	Laukai nepilni ir nevienodai patikimi, todėl griežtas filtras būtų per rizikingas.	Atliktos komponentų patikros, pogrupių analizė ir trūkstumų bandymai.
Patikimumas	Sistema gali atsakymo nepateikti, kai nėra pakankamai tikra.	Taip mažinama klaidų rizika sunkiausioms užklausoms.	Vertinta pagal tai, kiek užklausų sistema atsako, atkūrimą@5 ir rizikos santraukas.
Ko atsisakyta	Sudėtingesni sprendimai palikti tik kaip patikros, ne kaip pagrindas.	Dalis jų buvo lėtesni, mažiau aiškūs arba nepralenkė paprastesnio varianto.	Tikrintas mokomas perrikuotojas, kandidatų rinkinio dydis, kelių pavyzdžių režimas ir kiti vaizdo koduotojai.

5. Monetų atpažinimo eksperimento rezultatai

5.1. Tikslai ir tikrinamos prielaidos

Eksperimentų tikslas – patikrinti, ar pasirinkta paieškos schema iš tikro veikia praktiškai. Atskirai tikrinama paieška pagal vaizdą, papildomos informacijos perrikiavimas, patikimumas ir keli alternatyvūs variantai. Visi palyginimai daromi taip, kad kiekvieną svarbų sprendimą būtų galima įvertinti atskirai.

1. Paieška pagal vaizdo panašumą, naudojant požymių vektorius, suteikia stiprią bazę netolygiame uždavinyje, tačiau vien vaizdo informacijos nepakanka aukštam makro F1 lygiui, todėl reikalingi papildomi monetos laukai.
2. Papildomos informacijos perrikiavimas, naudojant svorį, skersmenį ir metų laikotarpį, reikšmingai pagerina galutinio atsakymo kokybę, o pagerėjimas turi matytis ir retų klasių grupėje, palyginti su prototipais ir kNN balsavimu vien iš požymių vektorių.
3. Metų laikotarpis turėtų labiausiai keisti kandidatų eilę, o skaitiniai laukai, tai yra svoris ir skersmuo, taip pat turėtų prisidėti, bet mažiau.
4. Kandidatų rinkinio didinimas turėtų pagerinti kokybę, pavyzdžiui, makro F1 ir atkūrimą@K, tačiau su vis mažesniu papildomu pagerėjimu. Tai turi parodyti aiškų kokybės ir skaičiavimo kainos kompromisą.
5. Taisyklė, kada sistema neturėtų pateikti atsakymo, paremta balų skirtumu tarp pirmos ir antros vietos. Ji turėtų gerinti rekomendacijų kokybę tada, kai iš anksto nustatyta mažiausia užklauso dalis, kuriai sistema turi pateikti atsakymą. Taip vertinamas atkūrimas@5 kartu su reikalavimu, kokios mažiausiai užklauso daliai turi būti pateiktas atsakymas.
6. Dėl papildomos informacijos triukšmo ir pasiskirstymų skirtumų mokomas modelis, apmokomas tik ant kandidatų, turėtų būti linkęs per daug prisitaikyti prie mokymo duomenų ir gali nusileisti perrikiavimui pagal paprastas taisykles.
7. Trūkstamas skersmuo turėtų išlikti vienu svarbiausių likusių klaidų šaltinių, todėl pogrupis be skersmens turėtų turėti aiškiai mažesnę makro F1. Jei tai pasitvirtins, šį ribojimą reikės aiškiai aptarti išvadose ir tolesniuose darbuose.

5.2. Vertinimo tvarka

Eksperimentuose naudojamas fiksuotas duomenų padalijimas į mokymo, validavimo ir testavimo aibes. Pagrindiniame duomenų variante, naudotame daugumai rezultatų, yra 5 634 mokymo pavyzdžiai, 1 882 validavimo pavyzdžiai ir 1 882 testavimo pavyzdžiai iš 45 klasių. Duomenų pasiskirstymas yra netolygus, todėl vien tik tikslumas (angl. accuracy) nėra pakankamas rodiklis.

Vaizdo požymių vektorių išgavimo dalis apmokoma tik mokymo aibėje, taikant lengvą vaizdų didinimą (augmentaciją), pavyzdžiui, rotacijas, poslinkius ir mastelio keitimą. Validavimo ir testavimo vertinimas atliekamas be vaizdų didinimo, kad rodikliai atspindėtų realių įvesčių kokybę.

Skaitinių papildomos informacijos laukų skalė suvienodinama taip, kad nebūtų duomenų nutekėjimo. Skalės suvienodinimo skaičiai apskaičiuojami tik mokymo aibėje ir tada be papildomo priderinimo taikomi validavimo bei testavimo aibėms.

Vertinimui naudojami penki rodikliai, visur pateikiami ta pačia seka, kad būtų lengva lyginti.

Pirma, tikslumas. Jis parodo bendrą teisingų sprendimų santykį. Antra, F1 pagal klasių dydį. Tai F1 rodiklis, kuriame klasės vertinamos pagal jų dydį, todėl jis dažniau atspindi dažnas klases. Trečia, makro F1. Tai vidurkis per klases, kiekvienai klasei suteikiant vienodą įtaką, todėl šis rodiklis yra pagrindinis netolygaus pasiskirstymo kontekste. Ketvirta, atkūrimas@K. Šis rodiklis parodo, ar teisinga klasė patenka į siūlomų kandidatų sąrašą, pavyzdžiui, kai $K = 5$. Penkta, patikimumas ir atvejai, kai sistema neturėtų pateikti atsakymo. Čia skaičiuojama, kiek užklausų sistema atsako, ir atitinkamas rodiklis, pavyzdžiui, atkūrimas@5, taikant skirtingus slenksčius. Pagrindiniam darbo rodikliui naudojama sąlyga pateikti atsakymą bent nustatyti atvejų daliai, todėl slenkstis parenkamas validacijos aibėje, o testavimo aibėje pateikiamas vieną kartą.

Papildomai, kad būtų matomas didelio netolygumo efektas, makro F1 pateikiamas ir trijose grupėse pagal mokymo aibėje esančių klasių pavyzdžių skaičių. Tai yra dažnos, vidutinio dažnio ir retos klasės, atitinkamai kai klasė turi bent 100, 20–99 arba 1–19 pavyzdžių. Tokios grupės leidžia įvertinti, ar pagerėjimas atsirado tik dažnoms klasėms, ar taip pat retoms.

Kad vertinimo tvarka būtų aiški ir patikrinama, kiekvienas rodiklis apibrėžiamas atskirai. Tikslumas parodo, kiek kartų sprendimas sutapo su teisinga etikete, tačiau didelio netolygumo sąlygomis jis dažnai labiau atspindi dažnas klases. F1 pagal klasių dydį baudžia už klaidas, bet vis tiek labiau remiasi dažnų klasių elgsena, todėl jis naudingas palyginimui, bet nepakankamas kaip pagrindinis kriterijus. Makro F1 šiame darbe yra pagrindinis rodiklis, nes jis kiekvienai klasei suteikia vienodą įtaką ir parodo, ar sistema tikrai pagerėjo retoms klasėms. Jo aiškinimas visada derinamas su grupėmis pagal dažnumą ir su klaidų analize. Atkūrimas@K parodo, kokios užklausų daliai teisinga klasė patenka į sąrašą. Todėl šis rodiklis tiesiogiai atitinka praktinį scenarijų, kai naudotojui pateikiamas kandidatų sąrašas. Rodiklis, kiek užklausų sistema atsako, parodo, kokios užklausų daliai sistema apskritai pateikia atsakymą pagal patikimumo slenkstį. Kartu su atkūrimu@K jis leidžia parodyti, kokia rekomendacijų kokybė pasiekama, jei sistema atsako tik daliai užklausų.

Tokia vertinimo tvarka tiesiogiai susieja literatūros analizėje aptartus motyvus su praktiniu vertinimu. Didelis netolygumas reikalauja makro F1 ir grupių. K geriausių kandidatų sprendimas reikalauja atkūrimo@K. Praktinis patikimumas reikalauja valdyti, kiek užklausų sistema atsako.

5.3. Eksperimentų palyginimo tvarka

Kad palyginimai būtų sąžiningi ir atkartojami, taikomos šios taisyklės. Visi variantai lyginami ta pačia vertinimo tvarka, tomis pačiomis duomenų dalimis ir tais pačiais pagrindiniais rodikliais.

Pirma, fiksuojamas duomenų padalijimas. Visi metodai lyginami tame pačiame mokymo, validavimo ir testavimo padalijime, jei nurodyta kitaip. Antra, nustatymai, pavyzdžiui, kandidatų rinkinio dydis, perrikiavimo svoriai ir balų skirtumo slenkstis, parenkami tik validacijoje, o testavimo aibėje pateikiami vienkartiniai rezultatai su užfiksuotais nustatymais. Trečia, visuose eksperimentuose pateikiamos tos pačios metrikos, o aiškinant rezultatus aiškiai atskiriamas vieno atsakymo režimas ir K geriausių kandidatų režimas. Ketvirta, kai tikrinamas konkretus modulis, laikoma fiksuota likusi sprendimo seka, kad būtų galima aiškiai priskirti rezultatų pokytį. Penkta, kur tik įmanoma, palyginimai atliekami ant tos pačios bazės, kad rezultatų skirtumai būtų priskirtini sprendimo taisyklėms, o ne atsitiktinumui. Vienintelė vieta, kur atsiranda daugiau atsitiktinumų, yra kelių pavyzdžių režimas, todėl ten rezultatai pateikiami kaip vidurkis ir standartinis nuokrypis per kelias sėklas.

Taip pat fiksuojama, kad kiekvienas rezultatas būtų lengvai patikrinamas. Kiekvienas pateiktas skaičius turi turėti ataskaitą su nustatymais, o kiekviena svarbi išvada turi turėti bent vieną papildomą patikrą, pavyzdžiui, grupę, komponentų patikrą, pogrupio analizę arba kokybinį pavyzdį. Tai padeda išvengti situacijos, kai diskusija remiasi vienu skaičiumi be konteksto ir tampa pažeidžiama dėl didelio netolygumo ar etikečių triukšmo.

Papildomai fiksuojama ir tyrimo logika. Pirma, kai lyginami metodai, pirmiausia tikrinama bazė, tai yra klasifikacija ir vien vaizdo paieška. Tada pridėjami moduliai, tai yra papildoma informacija ir patikimumas. Galiausiai pateikiami papildomi patikrinimai, tokie kaip mastelis, kelių pavyzdžių režimas, etikečių smulkumas ir klaidų analizė. Antra, rezultatai aiškinami ne tik per vieną skaičių, bet ir per dažnas, vidutinio dažnio ir retas klases, nes tai tiesiogiai atitinka darbo problemą ir tikslą.

5.4. Eksperimentai ir rezultatai

Eksperimentų dalis pradedama nuo bendros schemos ir pagrindinių rezultatų suvestinės. Joje atskiriama, kas tikrinta kaip galutinis sprendimo kelias, o kas buvo bandoma kaip alternatyva ir nepalikta galutiniame variante.

Kaip skaityti šią suvestinę. 5.1 pav. pirmoje dalyje parodyta, iš kokių duomenų pradėta, kokie vaizdo modeliai buvo lyginti ir kaip vaizdas paverčiamas skaičiais. 5.1 pav. antroje dalyje parodyta, kaip iš vaizdo sudaromas galimų atsakymų sąrašas, kaip jo tvarka keičiama pagal papildomą informaciją ir kaip sprendžiama, ar sistema turi pateikti atsakymą. 5.1 lentelėje pateikiama pagrindinių rezultatų suvestinė, kad būtų matyti ir galutinis variantas, ir bandomieji variantai, kurie naudos nedavė.

Pagrindinis eksperimentas	Bandomasis / tiriamasis	Galutinis pasirinktas sprendimas	Rezultatai / metrikos
---------------------------	-------------------------	----------------------------------	-----------------------

Duomenys		
Iš kur paimti duomenys ir kokie vaizdai naudoti Nomisma / MANTIS RDF įrašai, tai yra struktūruoti monetų aprašai Priekinės ir galinės pusės vaizdai Monetos nominalo klasė	Kokia papildoma informacija naudota Svoris Skersmuo Metų pradžia ir pabaiga Kai kuriuose bandymuose: valdovas, kalykla	Kaip duomenys paruošti Pašalinti įrašai be vaizdų Pašalinti dublikatai Paliktos klasės su bent 5 pavyzdžiais Sudarytos atskiros mokymo, validavimo ir testavimo aibės



Vaizdo modeliai	Vaizdo pavertimas skaičiais palyginimui
Lyginti vaizdo modeliai	Ką ši dalis daro

<p>Pagrindinis variantas yra MobileNetV2.</p> <p>Kiti palyginti modeliai buvo MobileNetV3Small, EfficientNetB0, EfficientNetV2B0 ir ResNet50V2.</p>	<p>Kiekvienas monetos vaizdas paverčiamas skaičių rinkiniu, kad būtų galima rasti panašiausius vaizdus.</p> <p>Pagrindiniam variantui pasirinktos abi monetos pusės.</p> <p>Papildomai lyginta tik priekinė pusė ir tik galinė pusė.</p>
---	--



Artimiausių panašių pavyzdžių paieška		
<p>Tiesioginis klasifikatorius</p> <p>Modelis iš karto parenka vieną klasę.</p> <p>Čia galimų atsakymų sąrašas nesudaromas.</p>	<p>Klasės centras</p> <p>Kiekviena klasė aprašoma vienu tipiniu atstovu.</p> <p>Užklausa lyginama su tais tipiniais atstovais.</p>	<p>kNN paieška</p> <p>Pirmiausia surandami panašiausi pavyzdžiai pagal vaizdą.</p> <p>Šis kelias pasirinktas pagrindiniam variantui.</p>
<p>Kiek variantų paliekama po pirmos paieškos</p> <p>K = 50</p> <p>K = 200</p> <p>K = 500, pasirinkta pagrindiniam variantui</p>	<p>Kas čia buvo tikrinta</p> <p>Ar užtenka vieno tipinio kiekvienos klasės atstovo, ar geriau remtis tikrais panašiais pavyzdžiais.</p>	<p>Kodėl tai svarbu</p> <p>Jei teisinga moneta po pirmo žingsnio nepatenka į sąrašą, vėliau jos vietos jau nebegalima pagerinti papildoma informacija.</p>

5.1 pav. (1 dalis). Duomenys, vaizdo modeliai ir vaizdo pavertimas skaičiais palyginimui

Galimų atsakymų tvarkos keitimas pagal papildomą informaciją	Sprendimas, ar sistema pakankamai tikra
<p>Kokia papildoma informacija naudota</p> <p>Svoris Skersmuo Metų pradžia ir pabaiga Žymė, kad lauko trūksta Kai kuriuose bandymuose: valdovas, kalykla</p>	<p>Pagal ką sprendta, ar sistema pakankamai tikra</p> <p>Skirtumas tarp 1 ir 2 vietos balų</p> <p>Pataisyta tikimybė, geriau atitinkanti realius rezultatus</p>
<p>Paprastos iš anksto nustatytos taisyklės</p> <p>Kandidatų perkėlimas aukštyne arba žemyn pagal papildomą informaciją</p> <p>Bauda už netinkamą laikotarpį</p>	<p>Ką leidžia šis žingsnis</p> <p>Atskirti atvejus, kai sistema atrodo pakankamai tikra, nuo atvejų, kai ji jau abejoja.</p>

<p>Lankstus artimiausių pavyzdžių skaičius</p> <p>Kandidatų sąrašo praplėtimas, kad jame nebūtų vien labai panašūs variantai</p> <p>Skirtingi būdai sujungti artimiausių pavyzdžių balsus</p> <p>Patikra, kaip rezultatą keičia skirtingas panašumo jautrumas</p>	<p>Nepateikti atsakymo, kai du geriausi variantai per arti vienas kito.</p> <p>Valdyti kompromisą tarp daugiau atsakytų užklausų ir mažesnės klaidų rizikos.</p>
<p>Sudėtingesni mokomi variantai, tirti tik kaip bandomieji</p> <p>Atskiras modelis kandidatų tvarkai keisti</p> <p>Modelis, tikrinantis užklausos ir kandidato porą</p> <p>FastAP, mokymas geresnei paieškos tvarkai</p> <p>ProxyNCA, mokymas su klasių atstovais</p> <p>PaCo tipo bandymas</p> <p>Užklausos praplėtimas pagal jau rastus panašius pavyzdžius</p> <p>Kandidatų tvarkos keitimas pagal platesnį kontekstą</p>	<p>Ką tikrino šis etapas</p> <p>Čia tikrinta, ar sistema turi atsakyti visada, ar tik tada, kai atsakymas atrodo pakankamai patikimas.</p> <p>Jei pirmas kandidatas tik vos lenkia antrą, sistema gali geriau atsakymo nepateikti.</p> <p>Taip mažėja klaidos rizika, nors atsakoma ne į visas užklausas.</p>



Galutinis pasirinktas variantas	Kaip vertinta	Kitos svarbios patikros
<p>Vaizdo modelis yra MobileNetV2.</p> <p>Paieškai naudota artimiausių panašių pavyzdžių paieška.</p> <p>Pradinis sąrašas turi 500 galimų atsakymų.</p> <p>Naudojamos abi monetos pusės.</p> <p>Antrame etape kandidatų tvarka keičiama pagal svorį, skersmenį ir metų laikotarpį.</p> <p>Patikimumo taisyklė leidžia nepateikti atsakymo, kai sistema nėra pakankamai tikra.</p>	<p>makro F1 rodo bendrą balą visoms klasėms.</p> <p>makro F1 retoms klasėms.</p> <p>atkūrimas@1 rodo, ar teisinga moneta pirma.</p> <p>atkūrimas@5 rodo, ar teisinga moneta tarp 5 variantų.</p> <p>atsakytų atvejų dalis.</p> <p>ECE rodo, ar tikimybės atitinka realų tikslumą.</p> <p>AURC rodo klaidų rizikos ir atsakytų atvejų dalies kompromisą.</p>	<p>Atskirų sprendimo dalių patikros.</p> <p>Palyginimas tarp įrašų, kur skersmuo žinomas ir nežinomas.</p> <p>Palyginimas tarp dažnų, vidutinių ir retų klasių.</p> <p>Bandymai, kai naujai klasei duodama tik keli pavyzdžiai.</p> <p>Klaidingų atvejų nagrinėjimas.</p>

5.1 pav. (2 dalis). Panašių monetų paieška, galimų atsakymų tvarkos keitimas ir galutinis sprendimas

5.1 lentelė. Pagrindinių eksperimentų rezultatų suvestinė

Variantas	Vieno atsakymo makro F1	makro F1 retoms klasėms	Išvada
Bazinis klasifikatorius (vienas atsakymas)	0,4258	0,3178	Nuo galutinio varianto atsilieka 0,3932 makro F1, o retoms klasėms skirtumas siekia 0,4796.

Prototipai (vien vaizdas)	0,4556	0,3790	Už galutinį variantą silpnesnis 0,3634 makro F1, o retoms klasėms skirtumas siekia 0,4184.
kNN (k=1, vien vaizdas)	0,4630	0,3687	Už galutinį variantą silpnesnis 0,3560 makro F1, todėl tinka tik kaip pirmas žingsnis.
kNN + perrikiavimas pagal papildomą informaciją (pool=200)	0,8096	—	Nuo galutinio varianto skiriasi tik 0,0094 makro F1, bet šiek tiek nusileidžia didesniai kandidatų sąrašui.
kNN + perrikiavimas pagal papildomą informaciją (pool=500)	0,8190	0,7974	Tai geriausias pagrindinės sistemos variantas šiame darbe. makro F1 siekia 0,8190, o retoms klasėms 0,7974.
Kandidatų sąrašo diversifikavimas (cap=3) + perrikiavimas	0,8221	—	Pagerėjimas prieš pagrindinį variantą siekia tik 0,0031 makro F1, todėl sistemos logikos nekeičia.
EfficientNetB0 + perrikiavimas	0,8258	—	Rezultatas 0,0068 geresnis už pagrindinį variantą, bet modelis lėtesnis.
MobileNetV2 palyginimui (vienoda vertinimo tvarka) + perrikiavimas	0,7220	—	Naudotas sąžiningam palyginimui, o rezultatas 0,0970 mažesnis už pagrindinę sistemos versiją.
Mokomas perrikiuotojas (LambdaRank, sąrašas=500)	0,7332	—	Nuo paprasto pool=500 varianto atsilieka 0,0858 makro F1, nors pats metodas sudėtingesnis.

Komponentų ir ribojimų patikros apibendrintos 5.2 lentelėje. Ji padeda atskirti, kurie pakeitimai pagerino sprendimą, o kurie tik padidino sudėtingumą.

5.2 lentelė. Komponentų ir ribojimų patikrų suvestinė

Patikra	makro F1	Išvada
Pilnas perrikiavimas (full, validacijoje)	0,7358	Atskaitos taškas, nes validacijoje pasiektas 0,7358 makro F1.
Be svorio, skersmens ir metų laukų (no_num, validacijoje)	0,6914	Be svorio, skersmens ir metų laukų makro F1 sumažėja nuo 0,7358 iki 0,6914, todėl jų nauda matoma, bet ribota.
Be pradžios ir pabaigos metų (no_date, validacijoje)	0,6311	makro F1 sumažėja iki 0,6311, todėl pradžios ir pabaigos metų laukai yra stipriausias papildomas signalas.

Skersmuo žinomas (testavimo aibėje)	0,8362	Kai skersmuo žinomas, makro F1 yra 0,1115 didesnis negu tada, kai jo nėra.
Skersmuo nežinomas (testavimo aibėje)	0,7247	Kai skersmuo nežinomas, makro F1 nukrenta iki 0,7247, todėl būtent šio lauko trūkumas labiausiai riboja kokybę.

5.3 lentelėje pateikiama patikimumo santrauka, kai sistema atsako bent į 0,8 dalį užklausų. Joje rodoma, kaip keičiasi atkūrimas@5, jei sistema atsako tik tada, kai yra pakankamai užtikrinta.

5.3 lentelė. Patikimumo rezultatų suvestinė, kai daliai atvejų sistema atsakymo nepateikia

Variantas	Kiek užklausų sistema atsako	atkūrimas@5	Ką tai duoda
Kai sistema visada pateikia atsakymą (pool=500, perikiavimas)	1,0000	0,9787	Atskaitos taškas, kai sistema visada pateikia atsakymą.
Kai atsakymas nepateikiamas pagal 1 ir 2 vietos balų skirtumą	0,8693	0,9768	Sistema atsako į 13,07 % mažiau užklausų, o atkūrimas@5 sumažėja tik 0,0019.
Kai atsakymas nepateikiamas pagal kalibruotą tikimybę (Platt)	0,8820	0,9813	Lyginant su 1 ir 2 vietos balų skirtumu, sistema atsako į 1,27 % daugiau užklausų, o atkūrimas@5 padidėja 0,0045.

5.4.1. Bazinis klasifikatorius

Pirmiausia tikrintas bazinis uždaro rinkinio sprendimas, tai yra softmax klasifikatorius, kuris pateikia vieną galutinį atsakymą. Ši patikra reikalinga kaip atskaitos taškas, parodantis, kiek galima pasiekti vien klasifikacijos požiūriu netolygiame monetų rinkinyje.

Bazinio klasifikatoriaus rezultatai pateikti 5.4 lentelėje. Naudotas tas pats apmokytas bazinis modelis ir tas pats duomenų variantas kaip paieškos sistemoje, tačiau atsakymas imtas tiesiai iš softmax išvesties. Rodikliai skaičiuoti atskirai validavimo ir testavimo aibėse.

5.4 lentelė. Bazinio klasifikatoriaus rezultatų suvestinė

Variantas	Tikslumas	F1 pagal klasių dydį	makro F1	Dažnos	Vidutinės	Retos
Softmax klasifikatorius	0,7784	0,7815	0,4258	0,7168	0,5028	0,3178

Bazinis klasifikatorius pasiekia priimtina bendrą tikslumą, tačiau makro F1 yra žemas, ypač retų klasių grupėje. Tai rodo, kad didelio netolygumo sąlygomis vieno atsakymo požiūris ir optimizavimas pagal bendrą vidurkį yra nepakankami, jei siekiama patikimai identifikuoti retas klases. Šis rezultatas motyvuoja pereiti prie K geriausių kandidatų paieškos scenarijaus ir ieškoti papildomo signalo atvejams, kai keli kandidatai atrodo panašiai.

Svarbu pastebėti, kad čia matomas tipinis didelio netolygumo efektas. Tikslumas ir F1 pagal klasių dydį gali atrodyti pakankami, tačiau makro F1 atskleidžia, kad klaidos koncentruojasi retose klasėse. Todėl šiame darbe bazinis klasifikatorius nėra laikomas konkurentu galutinei sistemai. Jis laikomas atskaitos tašku, parodančiu, ką duoda vieno atsakymo požiūris ir kokios yra jo ribos.

5.4.2. Paieška pagal vaizdo panašumą be papildomos informacijos

Toliau tikrinta, kiek vien vizualinė požymių vektorių erdvė leidžia spręsti monetų identifikavimą, kai sprendimas grindžiamas panašumo paieška. Čia lyginti du keliai. Tai buvo prototipų klasifikatorius ir kNN balsavimas iš artimiausių pavyzdžių.

Iš mokymo aibės pavyzdžių suformuojamas pavyzdžių rinkinys požymių vektorių erdvėje. Vertinami du sprendimai.

Paieškos pagal vaizdo panašumą rezultatai pateikti 5.5 lentelėje. Pirma, prototipai reiškia, kad kiekvienai klasei apskaičiuojamas požymių vektorių vidurkis, o prognozė gaunama pagal didžiausią panašumą prototipui. Antra, kNN balsavime prognozė gaunama iš artimiausių kaimynų.

5.5 lentelė. Paieškos pagal vaizdo panašumą ir susijusių patikrų rezultatų suvestinė

Kas tikrinta	Variantas	Rezultatai	Ką tai rodo
Ar kiekvieną klasę galima aprašyti vienu tipiniu pavyzdžiu vietoje visų panašių monetų	Prototipai	tikslumas 0,7418 makro F1 0,4556 retų klasių makro F1 0,3790	Šis būdas buvo silpnesnis už paiešką pagal panašiausius pavyzdžius.
Ar geriausia rinktis artimiausią panašų pavyzdį pagal vaizdą	kNN, k = 1	tikslumas 0,8225 makro F1 0,4630 retų klasių makro F1 0,3687	Bendras rezultatas geresnis, bet retos klasės vis dar atpažįstamos silpnai.
Ar kelių artimiausių pavyzdžių daugumos balsas pagerina atsakymą	majority	makro F1 0,7847	Rezultatas prastesnis už balų sumavimą pagal panašumą.
Ar artimesniems pavyzdžiams didesnę svorį duodantis balsavimas pagerina atsakymą	inv_distance	makro F1 0,7921	Rezultatas taip pat prastesnis už balų sumavimą pagal panašumą.
Ar panašumo balų sumavimas tarp artimiausių pavyzdžių duoda geriausią vieną atsakymą	score_softmax	makro F1 0,8190	Tai geriausia iš trijų paprastų balsavimo taisyklių.
Ar paieškai padeda tai, kad užklauso vaizdo aprašas papildomas artimiausių kaimynų vidurkiu	AQE	makro F1 0,7903 atkūrimas@5 0,9750	Pagrindinio rezultato nepagerino.
Ar kandidato vietą verta koreguoti pagal tai, kiek sutampa jo ir užklauso artimiausi kaimynai	Kontekstinis perrikiavimas	makro F1 0,7894 atkūrimas@5 0,9761	Pagrindinio rezultato nepagerino.
Ar verta riboti, kiek tos pačios klasės kandidatų gali likti sąrašo viršuje	Riba 3 vienai klasei	makro F1 0,8221	Davė nedidelį, bet nuoseklų pagerėjimą.

		atkūrimas@1 0,9671	
Ar didesnis pradinis kandidatų sąrašas išlaiko kandidatų įvairovės naudą	Pradinis sąrašas 1 000	makro F1 0,8190	Padidinus pradinį sąrašą pagerėjimas išnyko.
Ar FastAP mokymas, kuris modelį moko geriau sudėlioti kandidatų eilę, pagerina vaizdo paiešką	FastAP	makro F1 0,7335	Nepagerino pagrindinio kNN varianto.
Ar ProxyNCA mokymas, kai klasė aprašoma mokomu centru, pagerina vaizdo paiešką	ProxyNCA	makro F1 0,7572	Nepagerino pagrindinio kNN varianto.
Ar PaCo tipo mokymas padeda retoms klasėms vaizdo paieškoje	PaCo tipo variantas	makro F1 0,7656	Nepagerino pagrindinio kNN varianto.
Ar mokymas, kuris kartu artina panašias monetas ir moko klasę, pagerina galutinį sprendimą	SupCon+CE	makro F1 0,7632 be perrikiavimo 0,5143	Galutinio sprendimo nepagerino.

Vien paieška pagal vaizdo panašumą pagerina bendrą tikslumą, bet retų klasių problemos neišsprendžia. makro F1 lieka žema, ypač retoms klasėms. Tai rodo, kad vien vaizdo neužtenka ir reikia dar vieno signalo, kuris padėtų atskirti artimus atvejus.

kNN su $k = 1$ yra labai stiprus dažnoms klasėms, nes jos pavyzdžių rinkinyje turi daugiau artimų pavyzdžių. Dėl to artimiausio kaimyno taisyklė dažniau pataiko būtent ten. Tačiau ši nauda nepersikelia į retas klases, nes joms trūksta pakankamai gerų pavyzdžių, o vizualų panašumą dažnai užgožia dažnesnės klasės. Būtent tokioje vietoje papildoma informacija gali duoti ypač daug naudos.

Papildoma patikra su keliomis paprastomis paieškos taisyklėmis parodė, kad ne visos tokios taisyklės yra naudingos šiame scenarijuje.

Kad būtų patikrinta, ar perrikiavimą galima išmokti, o ne užrašyti ranka, įjungtas mokomas modelis, kuris po paieškos keičia kandidatų tvarką. Jis kiekvienai užklauso ir kandidato porai prognozuoja, kiek kandidatas panašus į teisingą atsakymą. Naudoti paprasti ir paaiškinami požymiai, pavyzdžiui, kosinusinis panašumas, skaičiais pateiktų laukų atstumas, metų sutapimas, autoriteto sutapimas ir kandidato vieta sąrašė. Mokymo metu naudoti ir lengvi, ir sunkūs neigiami pavyzdžiai. Vis dėlto rezultatas buvo neigiamas. Nors kai kurie tarpiniai rodikliai atrodė geri, galutinio atsakymo makro F1 buvo žemesnis negu bazėje. Tai parodo, kad net ir toks mokomas modelis čia gali būti per nestabilus, jei neturi daugiau ir tikslesnės informacijos apie kandidatus arba geriau pritaikyto mokymo tikslo (žr. 1 priedo 1.7 poskyrį).

Papildomai buvo patikrinti keli kiti požymių vektorių mokymo variantai ir keli vaizdo modeliai. Šios patikros skirtos ne naujam pagrindiniam sprendimui sukurti, o patikrinti, ar vien pakeitus vaizdo modelį arba mokymo tikslą galima pagerinti paiešką pagal vaizdo panašumą.

5.4.2.1. Požymių vektorių išgavimo modelių palyginimas

Atskirai sulyginta, ar vien pakeitus požymių vektorių išgavimo modelį galima pagerinti paiešką pagal vaizdo panašumą ir galutinį rezultatą po papildomos informacijos perrikiavimo.

Buvo apmokyti keli alternatyvūs požymių vektorių išgavimo modeliai, laikant tą patį duomenų variantą ir tą pačią vertinimo tvarką. Tikrinti MobileNetV2 [31], MobileNetV3Small [32],

EfficientNetB0 [33], EfficientNetV2B0 [34] ir ResNet50V2 [35]. Kiekvienam modeliui vertinti du variantai. Pirma, kNN su $k = 1$ vien pagal vaizdo panašumą. Antra, tas pats kNN su papildomos informacijos perrikiavimu, naudojant 500 kandidatų sąrašą, vienodus svorius svoriui, skersmeniui ir metų informacijai, metų intervalo ribojimą ir k reikšmę, kuri keičiasi tada, kai sprendimas nėra aiškus.

Požymių vektorių išgavimo modelių palyginimas pateiktas 5.6 lentelėje. Iš patikrintų architektūrų kontroliniame palyginime geriausiai pasirodė EfficientNetB0. Vis dėlto pagrindinėje sistemos versijoje paliktas MobileNetV2, nes jis greitesnis, o visas sprendimas su juo išliko pakankamai stiprus.

5.6 lentelė. Požymių vektorių išgavimo modelių palyginimo suvestinė

Koduotojas	Vieno atsakymo makro F1	Paieška + perrikiavimas makro F1
EfficientNetB0	0,4759	0,8258
EfficientNetV2B0	0,4542	0,7536
MobileNetV3Small	0,2535	0,6727
ResNet50V2	0,4679	0,7399
MobileNetV2 pagrindinis vaizdo modelis sweep (vienoda vertinimo tvarka)	0,4403	0,7220
Pagrindinė sistemos versija (MobileNetV2)	0,4630	0,8190

Ši patikra parodė, kad vien pakeisti modelio architektūrą neužtenka. Geresnė pradinė paieška gali padėti, bet galutinis rezultatas vis tiek priklauso nuo to, kokius požymių vektorius sukuria modelis. Palyginimas atliktas tuo pačiu duomenų variantu ir tuo pačiu perrikiavimo režimu, todėl skirtumai čia siejami būtent su pasirinktu vaizdo modeliu.

Svarbu atskirti dvi MobileNetV2 eilutes. Palyginimui skirta MobileNetV2 eilutė gauta modelį apmokius iš naujo kaip šio palyginimo dalį, laikant tas pačias mokymo ir vertinimo sąlygas kaip ir kitiems modeliams. Tai nėra tas pats išsaugotas modelis, kuris naudotas pagrindinėje sistemos versijoje viso darbo eksperimentuose. Dėl mokymo atsitiktinumo ir dėl to, kad šiame palyginime nebuvo daroma ilga nustatymų paieška, iš naujo apmokytas MobileNetV2 gavo prastesnį rezultatą. Dėl to lentelėje paliktos abi eilutės. Viena rodo sąžiningą skirtingų modelių palyginimą vienodu režimu, o kita rodo galutinės sistemos rezultatą. Jei tikslas būtų vien tik maksimalus rezultatas šiame palyginime, geriausiai pasirodė EfficientNetB0, tačiau jis yra lėtesnis, todėl šiame darbe paliktas tik kaip papildoma patikra.

Vien vaizdo paieška jau duoda apčiuopiamą naudą, bet viena pati dar nepakelia retų klasių rezultatų pakankamai aukštai. Dėl to šiame darbe ji palikta kaip būtinas pirmas žingsnis, o ne kaip pilnas sprendimas.

5.4.3. Papildomos informacijos perrikiavimo patikros

Šioje dalyje tikrinta, kiek naudos duoda papildomos informacijos perrikiavimas ir kurie jo komponentai labiausiai prisideda prie rezultato.

Naudojamas tas pats požymių vektorių paieškos pagrindas kaip Eksperimente 2, tačiau po pradinės atrankos kandidatai perrikiuojami pagal papildomos informacijos atitikimą. Pagrindiniame variante naudoti tokie nustatymai.

Pagrindiniame variante naudotas 500 kandidatų sąrašas. Svoriumi, skersmeniui ir metų informacijai duotas vienodas svoris po 0,5, o vaizdo signalui paliktas svoris 1,0. Taip pat įjungtas metų intervalo ribojimas ir k reikšmės keitimas tada, kai sprendimas nėra aiškus.

5.7 lentelėje pateikta komponentų patikra, kurioje validacijoje su fiksuotais svoriais išjungiami atskiri papildomos informacijos terminai, pavyzdžiui, no_num arba no_date.

5.7 lentelė. Papildomos informacijos perrikiavimo patikrų rezultatų suvestinė

Patikra	Variantas	Rezultatai	Ką tai rodo
Pagrindinis variantas	Pilnas perrikiavimas	tikslumas 0,9516 F1 pagal klasių dydį 0,9492 makro F1 0,8190 atkūrimas@1 0,9660	Pagrindinis darbo variantas.
Komponentų patikra	Pilnas variantas validacijoje	makro F1 0,7358	Atspirties taškas komponentų palyginimui.
Komponentų patikra	Be svorio, skersmens ir metų laukų	makro F1 0,6914	Rezultatas krenta.
Komponentų patikra	Be datos	makro F1 0,6311	Data svarbiausia.
Pograpių analizė	Skersmuo žinomas	makro F1 0,8362	Pilnesni duomenys padeda.
Pograpių analizė	Skersmuo nežinomas	makro F1 0,7247	Trūkstamas skersmuo aiškiai kenkia.
Kategorinis metaduomuo	Be monetos leidybos vietos signalo	makro F1 0,7745	Šio patikrinimo bazė.
Kategorinis metaduomuo	Mažas monetos leidybos vietos bonusas	makro F1 0,7932	Gali padėti kaip lengvas papildomas balas.
Kategorinis metaduomuo	Griežtas atmetimas	makro F1 0,7813	Griežtas filtras žalingas.
Viena ar dvi pusės	Tik priekinė pusė	makro F1 0,7426	Vienos pusės neužtenka.
Viena ar dvi pusės	Tik galinė pusė	makro F1 0,7520	Vienos pusės neužtenka.
Viena ar dvi pusės	Abi pusės	makro F1 0,8190	Stabiliausias variantas.

Pusių svėrimas	$\alpha = 0,2$	makro F1 0,7806	Blogiau už bazę.
Pusių svėrimas	$\alpha = 0,5$	makro F1 0,8087	Blogiau už bazę.
Pusių svėrimas	$\alpha = 0,8$	makro F1 0,7974	Blogiau už bazę.
Sąrašų vietų sujungimas	RRF, $k_0 = 60$	makro F1 0,6479	Žalingas.
Trūkstumų patikra	$p = 0,1$	validacijos makro F1 0,7165	Nepilna informacija mažina naudą.
Trūkstumų patikra	$p = 0,2$	validacijos makro F1 0,7129	Nauda toliau mažėja.
Trūkstumų patikra	$p = 0,3$	validacijos makro F1 0,7021	Didelis trūkstumas aiškiai kenkia.
Papildomos alternatyvos	Klasės lygmens terminas	makro F1 0,7816	Bazės neaplenkė.
Papildomos alternatyvos	Softmax išankstinis įvertis	makro F1 0,7376	Bazės neaplenkė.

Lentelė rodo, kad didžiausią naudą šiame bloke duoda metų intervalo sutapimas. Svoris ir skersmuo taip pat padeda, bet jų nauda mažesnė ir labiau priklauso nuo to, ar tie laukai yra užpildyti.

Todėl trūkstamas skersmuo lieka vienu svarbiausių šio darbo ribojimų, o griežti filtrai ar papildomi statistiniai terminai bazinio sprendimo nepagerino.

Papildomas kategorinis laukas

Toliau tikrinta, ar papildomas kategorinis laukas, tai yra monetos leidybos vieta, gali pagerinti sąrašą netolygiame scenarijuje ir ar tokį signalą saugu įtraukti neišmetant teisingo kandidato dėl triukšmingų ar nepilnų įrašų.

Šiam patikrinimui naudotas atskiras duomenų variantas su monetos leidybos vietos lauku. Šis laukas įvestas kaip papildoma perrikiavimo dalis, kuri duoda premiją kandidatams su sutampančia monetos leidybos vieta, bet jų neišmeta. Papildomai tikrintas ir griežtas variantas, kai kandidatai be sutapimo visai pašalinami.

Monetos leidybos vietos laukas šiame variante buvo naudingas tik tada, kai jis naudotas kaip nedidelis papildomas balas, o ne kaip atmetimo taisyklė.

Vienos pusės ir abiejų pusių palyginimas

Čia sulyginta, ar galutinei sistemos kokybei tikrai reikia abiejų monetos pusių, ar pakaktų tik vienos. Šis patikrinimas tiesiogiai parodo, ar priekinės ir galinės pusės jungimas yra būtina sprendimo dalis, ar tik papildomas patogumas.

Naudota ta pati bazinė konfigūracija su papildomos informacijos perrikiavimu kaip Eksperimente 3. Keista tik vaizdo informacija. Vietoje abiejų pusių naudota arba tik priekinė pusė, arba tik galinė pusė. Iš kiekvienos pusės imtas vienas 1 280 ilgio požymių vektorius.

Šis palyginimas rodo, kad viena pusė dažnai pakanka tam, kad teisinga moneta patektų į kandidatų sąrašą, bet vien jos nepakanka stabiliai išskirti teisingą atsakymą į pirmą vietą. Todėl šiame darbe

priekinė ir galinė pusė naudojamos kartu. Taip gaunamas stabilesnis sprendimas, kai reikia atskirti labai panašias monetas.

Pusių sujungimo svorio paieška

Po to tikrinta, ar priekinė ir galinė pusė sprendimui padeda nevienodai ir ar galima pagerinti sprendimą vienai pusei suteikiant didesnę svorį. Tai natūralus tęsinys po Eksperimento 3.1.

Naudojama ta pati bazinė konfigūracija su papildomos informacijos perrikiavimu kaip Eksperimente 3, tai yra 500 kandidatų sąrašas, perrikiavimas pagal papildomą informaciją, metų intervalo ribojimas ir k reikšmė, kuri keičiasi tada, kai sprendimas nėra aiškus. Skiriasi tik priekinės ir galinės pusės požymių vektorių sujungimas. Vietoje fiksuotos dviejųusių jungties taikomas sujungimo svėrimas su α reikšme. Patikrintos trys reikšmės, 0,2, 0,5 ir 0,8.

Šiame duomenų variante fiksuotas dviejųusių sprendimas pasirodė stabilesnis negu paprastas sujungimo svėrimas. Tai rodo, kad abi monetas pusės viena kitą papildo, bet nuolatinis vienos pusės per didelis svoris dažniau kenkia, negu padeda. Labiausiai tai matyti tada, kai reikia atskirti labai artimas denominacijas. Dėl to sujungimo svėrimas paliktas tik kaip tolimesnė kryptis. Prie jo būtų verta grįžti tik turint papildomą kokybės signalą apie kiekvieną pusę.

Kandidatų vietomis paremtas papildomos informacijos sujungimas

Dar vienoje patikroje išbandyta literatūroje minima idėja jungti kandidatus pagal jų vietas skirtinguose sąrašuose. Vietoje balų sumavimo žiūrėta, kelintoje vietoje kandidatas yra kiekviename sąraše, ir iš šių vietų sudaroma galutinė tvarka. Taip bandoma sumažinti jautrumą skirtingoms balų skalėms. Šiame darbe tai įgyvendinta kaip Reciprocal Rank Fusion, arba RRF.

Naudojama ta pati bazinė konfigūracija su papildomos informacijos perrikiavimu kaip Eksperimente 3, tai yra 500 kandidatų sąrašas, perrikiavimas pagal papildomą informaciją, metų intervalo ribojimas ir k reikšmė, kuri keičiasi tada, kai sprendimas nėra aiškus. Skiriasi tik galutinio sąrašo skaičiavimas. Vietoje balų sumos taikomas RRF su k_0 reikšme 60.

Kandidatų vietomis paremtas sujungimas šiame scenarijuje pasirodė žalingas. Smulkaus detalumo denominacijų atskyrimui svarbu ne tik tai, kuris kandidatas yra aukščiau sąraše, bet ir kiek jo balas skiriasi nuo kitų kandidatų balų. RRF šį balų skirtumą paslepia, todėl gali iškreipti sąrašo viršų. Dėl to šiame darbe pagrindiniu metodu paliktas balų sumavimas su validacijoje parinktais svoriais (žr. 1 priedo 1.8 poskyrį).

Papildomos informacijos triukšmo atsparumas

Atskira patikra imitavo situaciją, kai papildoma informacija tampa labiau nepilna. Čia žiūrėta, kaip kinta sprendimo patikimumas ir validacijos rezultatai, nes realiose užklausoje dalis papildomos informacijos gali būti nepateikta.

Naudojama bazinė konfigūracija su papildomos informacijos perrikiavimu ir 500 kandidatų sąrašu, paliekant tuos pačius perrikiavimo svorius. Validavimo aibėje dirbtinai užmaskuojama dalis skersmens ir metų laukų, o testavimo aibė paliekama nepakeista. Patikrintos trys trūkstumumo tikimybės, 0,1, 0,2 ir 0,3.

Rezultatai rodo, kad sprendimas remiasi papildomos informacijos prieinamumu ir kad papildomos informacijos nepilnumas yra realus rizikos šaltinis renkant nustatymus validacijoje. Tai sustiprina sprendimą naudoti papildomos informacijos perrikiavimą be griežto atmetimo ir papildomai rodyti pogrupius pagal tai, ar papildoma informacija pateikta.

Papildomos informacijos naudojimas kandidatams atmesti

Toliau tikrinta, ar papildomą informaciją verta naudoti kaip kandidatų atmetimo taisyklę, o ne tik kaip būdą pakeisti kandidatų vietas sąrašė. Taip buvo žiūrima, ar griežtesnis filtravimas pagerina rezultata, ar tik padidina riziką per anksti pašalinti teisingą kandidatą.

Griežtas kandidatų atmetimas šiame rinkinyje pasirodė per rizikingas. Kai papildoma informacija nepilna arba netiksli, teisingas kandidatas gali būti pašalintas per anksti. Dėl to šiame darbe papildoma informacija palikta tik kandidatų tvarkai pakeisti.

Papildomos informacijos perrikiavimo alternatyvos

Galiausiai šiame bloke buvo tikrintos trys alternatyvios rikiavimo korekcijos. Pirma, tikrinta klasės lygmens papildomos informacijos korekcija. Ji prideda balą tada, kai kandidato papildoma informacija panaši į tai, kas dažnai pasitaiko jo klasėje. Antra, tikrintas vieno atsakymo klasifikatoriaus balo priedas. Jis padidina kandidato balą tada, kai atskiras klasifikatorius labiau pasitiki tuo kandidatu. Trečia, tikrintas trūkstumų reikšmių suvienodinimas. Šiame variante dalis neaiškių papildomos informacijos reikšmių laikomos ta pačia nežinoma reikšme. Šie bandymai parodo, ar tokios korekcijos iš tikro pagerina sistemą, ar tik sustiprina triukšmą, kai papildoma informacija netiksli arba nepilna.

Naudota ta pati bazinė konfigūracija su 500 kandidatų sąrašu ir tais pačiais papildomos informacijos svoriais. Klasės lygmens korekcija tikrinta su keliomis svorio reikšmėmis. Vieno atsakymo klasifikatoriaus balo priedas tikrintas su keliomis lambda reikšmėmis, tai yra su skirtingu šio priedo stiprumu. Reikšmių suvienodinimas buvo vertintas kaip atskiras taip arba ne variantas, kai dalis reikšmių laikomos lygiavertėmis trūkstumoms.

Šie bandomieji variantai parodė, kad papildomi pradiniai įverčiai ir klasės lygmens papildomos informacijos korekcijos šiame monetų scenarijuje savaime nepadeda. Tikėtina priežastis yra papildomos informacijos triukšmas ir nepilnumas, ypač kai metų intervalai pateikiami kaip pradžios ir pabaigos metai. Prie to prisideda ir netolygus klasių pasiskirstymas. Tokiose sąlygose tokios korekcijos gali per stipriai sustiprinti dažnas klases arba netikslius dėsningumus. Dėl to šiame darbe jos paliktos išjungtos, o papildoma informacija naudojama tik per aiškius sutapimo balus su svoriais.

Komponentų patikros parodė, kad ne visi papildomi laukai duoda vienodą naudą. Svarbiausia buvo metų laikotarpio informacija. Skaitiniai laukai taip pat padėjo, bet jų nauda mažesnė ir labiau priklauso nuo to, ar laukai užpildyti. Todėl galutiniame variante paliktas tik tas papildomos informacijos derinys, kuris aiškiai pagerina rezultata.

5.4.4. Kandidatų rinkinio dydžio įtaka

Šioje dalyje tikrinta, kaip kandidatų sąrašo dydis veikia sprendimo kokybę ir ar papildomas skaičiavimo kaštas duoda apčiuopiamą naudą.

5.8 lentelėje lyginamos dvi to paties perrikiavimo logikos konfigūracijos, keičiant tik kandidatų sąrašo dydį. Naudoti du variantai, 200 ir 500 kandidatų.

5.8 lentelė. Kandidatų sąrašo dydžio, retinimo ir vaizdų apdorojimo patikrų suvestinė

Patikra	Variantas	Rezultatai	Ką tai rodo
Kandidatų sąrašo dydis	100 kandidatų	makro F1 0,7981	Per mažas sąrašas.
Kandidatų sąrašo dydis	200 kandidatų	tikslumas 0,9479 makro F1 0,8096 atkūrimas@5 0,9793	Mažesnė kaina, bet silpnesnė vieno atsakymo kokybė.
Kandidatų sąrašo dydis	500 kandidatų	tikslumas 0,9516 makro F1 0,8190 atkūrimas@5 0,9787	Bazinis kompromisas.
Kandidatų sąrašo dydis	1 000 kandidatų	makro F1 0,8208	Nauda maža.
Pavyzdžių rinkinio retinimas	Centroidas, paliekant 50 %	makro F1 0,6868	Per daug nupjauna įvairovę.
Pavyzdžių rinkinio retinimas	Atsitiktinai, 50 %, sėkla 0	makro F1 0,7326	Geriau už centroidą, bet aiškiai silpniau už bazę.
Pavyzdžių rinkinio retinimas	Atsitiktinai, 75 %, sėkla 0	makro F1 0,6970	Nestabilus.
Pavyzdžių rinkinio retinimas	Atsitiktinai, 75 %, sėkla 1	makro F1 0,7821	Nestabilus.
Dažnų klasių riba	max 200	makro F1 0,8056 mokymo rinkinys 2 002	Geriausias retinimo kompromisas.
Dažnų klasių riba	max 100	makro F1 0,7895 mokymo rinkinys 1 316	Daugiau taupo, bet labiau kainuoja kokybę.
Vaizdų apdorojimas	CLAHE	makro F1 0,7744	Neigiamas poveikis.
Vaizdų apdorojimas	Denoise + CLAHE	makro F1 0,7342	Dar blogiau.
Vaizdų apdorojimas	Fono šalinimas	makro F1 0,2617	Labai žalingas.

Kandidatų sąrašo didinimas nuo 200 iki 500 pagerino makro F1 nedaug, bet ta pačia kryptimi. Atkūrimas@5 abiem atvejais liko labai aukštas, todėl didesnis sąrašas čia labiau padeda surinkti jau rastus kandidatus, o ne tik tada įtraukia teisingą klasę į sąrašą. Praktikoje 200 kandidatų gali užtekti, kai svarbiausias greitis, o 500 kandidatų verta rinktis, kai priimtina didesnė skaičiavimo kaina dėl geresnio vieno atsakymo rezultato.

Papildoma patikra patvirtino, kad net su 200 kandidatų teisinga klasė į kandidatų sąrašą patenka beveik visada. Testavimo hit_rate buvo apie 0,993, o su 500 kandidatų jis pagerėjo tik nežymiai iki maždaug 0,996. Tai reiškia, kad sąrašo didinimo nauda daugiausia atsiranda kandidatų eilės viduje.

Atkūrimas@5 beveik nesikeitė, bet makro F1 didėjo. Todėl didesnis kandidatų sąrašas dažniau padėjo teisingą kandidatą perkelti į pirmą vietą, o ne pakeitė patį faktą, ar teisingas kandidatas buvo tarp K geriausių. Tai atitinka sprendimo logiką, kur papildomi atributai pirmiausia taiso kandidatų eilę jau sudarytame sąraše.

5.4.4.1. Pavyzdžių rinkinio retinimas

Toliau tikrinta, ar galima sumažinti mokymo aibės pavyzdžių rinkinį ir taip sumažinti paieškos kainą neprarandant kandidatų sąrašo kokybės ir vieno atsakymo kokybės. Jei didesnis kandidatų sąrašas duoda naudą, tada kyla klausimas, ar galima mažinti pavyzdžių rinkinį, bet išlaikyti panašų atkūrimą@K.

Naudojama ta pati bazinės sistemos konfigūracija kaip Eksperimente 4 su 500 kandidatų sąrašu. Retinamas tik mokymo aibės pavyzdžių rinkinys, o validavimo ir testavimo aibės paliekamos nepakitusios. Patikrintos trys paprastos pavyzdžių atrankos taisyklės.

Mastelio bandymai parodė, kad skaičiavimo kainą mažinti reikia atsargiai. Agresyvus pavyzdžių retinimas ir papildomas vaizdų apdorojimas dažniausiai blogino rezultatą. Geresnis kompromisas atsirado didinant kandidatų sąrašą arba ribojant, kiek dažnų klasių pavyzdžių gali kartotis sąrašo viršuje.

Dažnų klasių ribojimas geriau atitiko netolygaus rinkinio problemą. Jis mažino dažnų klasių perteklinį kartojimąsi kandidatų sąraše, bet nekarpė retų klasių pavyzdžių. Todėl pavyzdžių rinkinį buvo galima sumažinti su mažesne kokybės kaina. Jei rinkinys augtų, pirmiausia verta riboti per didelę dažnų klasių įtaką, o ne vienodai retinti visas klases.

5.4.4.2. Vaizdų pirminio apdorojimo patikros

Atskira patikra tikrino, ar paprasti vaizdo paruošimo žingsniai pagerina požymių vektorių paiešką netolygiame monetų rinkinyje. Svarbiausias klausimas buvo ne tai, ar vaizdas atrodo švaresnis, o ar po tokio apdorojimo geriau randamos teisingos klasės.

Naudoti tie patys baziniai nustatymai su papildomos informacijos perrikiavimu kaip Eksperimente 4, tai yra 500 kandidatų sąrašas ir tie patys perrikiavimo svoriai. Keistas tik vaizdų paruošimas prieš požymių vektorių išgavimą. Tikrinti trys variantai. Tai buvo CLAHE kontrasto pagerinimas, denoise kartu su CLAHE ir fono šalinimas.

Vaizdo paruošimo bandymai parodė, kad smulkaus detalumo monetų uždavinyje švaresnis vaizdas nebūtinai duoda geresnę paiešką. Kai apdorojimas sukuria nenatūralias tekstūras arba nutrina tekstą ir simbolius, rezultatas prastėja. Todėl galutinis pagerėjimas šiame darbe siejamas ne su vaizdo valymu, o su paieškos ir papildomų atributų perrikiavimo seka.

Didesnis kandidatų sąrašas iš tikro padeda, bet tik iki ribos. Dėl to galutiniame variante pasirinktas ne didžiausias galimas sąrašas, o toks dydis, kuris dar duoda naudos, bet jau neapkrauna sistemos be reikalo.

5.4.5. Patikimumas ir kada sistema neturėtų pateikti atsakymo

Šioje dalyje tikrinta, ar taisyklė, kada sistema neturėtų pateikti atsakymo, leidžia pagerinti rekomendacijų kokybę, kai iš anksto nustatoma, kad sistema turi atsakyti bent į tam tikrą dalį atvejų. Kartu buvo ieškoma vieno aiškaus rodiklio, kuris sujungtų K geriausių kandidatų naudą ir patikimumo klausimą. Šie bandymai atlikti su pasirinktu vaizdo koduotoju MobileNetV2, o kelių vaizdo koduotojų palyginimas pateiktas Eksperimente 2.

5.9 lentelėje pateikiami patikimumo ir neatsakymo patikrų rezultatai. Riba, nuo kurios sistema neturėtų pateikti atsakymo, parenkama validavimo aibėje taip, kad atkūrimas@5 būtų kuo didesnis, tačiau kartu sistema turi atsakyti bent į 0,8 dalį užklausų. Tada ta pati riba vieną kartą pritaikoma testavimo aibei.

5.9 lentelė. Patikimumo ir neatsakymo patikrų rezultatų suvestinė

Patikra	Variantas	Rezultatai	Ką tai rodo
Balų skirtumo slenkstis	0,005	makro F1 0,8205 atkūrimas@5 0,9768 atsakytų atvejų dalis 0,8693	Subalansuotas variantas.
Balų skirtumo slenkstis	Griežtesnis slenkstis	makro F1 0,8254 atkūrimas@5 0,9755 atsakytų atvejų dalis 0,7582	Kokybė kyla, bet sistema atsako rečiau.
atkūrimas@K be atmetimų	K = 1	0,9485	Pirmos vietos tikslumas.
atkūrimas@K be atmetimų	K = 3	0,9734	Keli kandidatai greitai pakelia naudingumą.
atkūrimas@K be atmetimų	K = 5	0,9787	Pagrindinis darbo rodiklis.
atkūrimas@K be atmetimų	K = 10	0,9857	Toliau gerėja lėčiau.
atkūrimas@K be atmetimų	K = 20	0,9883	Papildoma nauda jau maža.
Atsakytų atvejų dalis pagal grupes	Dažnos klasės	0,8621	Ne mažiausia grupė.
Atsakytų atvejų dalis pagal grupes	Vidutinio dažnio klasės	0,9314	Atsakoma dažniausiai.
Atsakytų atvejų dalis pagal grupes	Retos klasės	0,9565	Ši taisyklė retų klasių neslopino.
Kalibruota tikimybė	Validacijoje parinktas slenkstis	makro F1 0,7775 atkūrimas@5 0,9813 atsakytų atvejų dalis 0,8820	Geriau rikiuoja saugius atvejus, bet blogina vieno atsakymo kokybę.

Kalibruota tikimybė	0,95 tikimybės slenkstis	atsakytų atvejų dalis dažnoms klasėms 0,8919 atsakytų atvejų dalis vidutinėms klasėms 0,7843 atsakytų atvejų dalis retoms klasėms 0,7826	Labiau mažina tai, kokiai daliai vidutinių ir retų klasių sistema atsako.
---------------------	--------------------------	--	---

Papildomai, kad būtų aišku, ką reiškia K geriausių kandidatų rekomendacijos, kai sistema visada pateikia atsakymą, po papildomos informacijos perrikiavimo buvo įvertintas atkūrimas@K kelioms K reikšmėms, be atmetimų, tai yra kai sistema atsako į visas užklausas.

Papildomai buvo tikrinta, kaip slenkstį veikia pasirinktas rodiklis. Vietoje atkūrimo@5 bandyta slenkstį rinkti pagal makro F1, vis tiek laikantis sąlygos, kad sistema turi atsakyti bent į 0,8 dalį užklausų. Šiame darbe naudotame slenksčių tinkle galutinis pasirinkimas sutapo, nes šią sąlygą tenkino tik dvi reikšmės. Praktikoje tai reiškia, kad sprendimą čia nulėmė ne vien pasirinktas rodiklis, bet ir minimalus priimtinas atsakymų kiekis. Tai sustiprina išvadą, kad patikimumo mechanizme svarbi ne tik optimizuojama kokybė, bet ir aiški riba, kiek užklausų sistema turi atsakyti.

Papildomai buvo tikrinta, ar patikimumo signalą galima paversti lengvai suprantama tikimybe. Tam naudotas Platto kalibravimas (angl. Platt scaling), tai yra logistinė regresija. Balų skirtumas tarp pirmos ir antros vietos buvo paverstas į tikimybę, kad sprendimas teisingas. Toliau tikrinta, ar ši tikimybė iš tikro atitinka realų tikslumą. Tikimybės kokybė tikrinta trimis rodikliais. Brier balas rodo, kiek prognozuota tikimybė skiriasi nuo realaus atsakymo teisingumo. ECE rodo vidutinę tikimybų paklaidą, o MCE rodo didžiausią tikimybų paklaidą.

Balų skirtumas tarp pirmos ir antros vietos vertintas ne tik vienu slenksčiu, bet ir per visą rizikos kreivę ir atsakytų užklausų dalį. Šiame darbe rizika apibrėžta kaip klaidų dalis tarp atsakytų užklausų, o atsakytų užklausų dalis parodo, kokiai užklausų daliai sistema nusprendžia atsakyti. Toks vertinimas parodo, ar balų skirtumas gerai atskiria paprastesnius ir sunkesnius atvejus.

Rezultatai parodė, kad kalibruota tikimybė užklausas rikiuoja geriau negu paprastas balų skirtumas. Pavyzdžiui, pasirinkus maždaug 80 % labiausiai užtikrintų užklausų, kalibruota tikimybė davė didesnę atkūrimą@5 ir mažesnę riziką. Tai reiškia, kad kalibravimas čia yra ne vien papildomas rodiklis. Jis praktiškai padeda nuspręsti, kuriems atvejams atsakymą verta pateikti pirmiausia.

Taisyklė, kada sistema neturėtų pateikti atsakymo, pagerino rekomendacijas tarp tų užklausų, kurioms atsakymas vis dar pateikiamas, ir kartu išlaikė atsakymą maždaug 0,87 užklausų daliai. Tai reiškia, kad sistema gali atsakyti tada, kai pirmi kandidatai pakankamai skiriasi, o rizikingesnius atvejus palikti papildomai patikrinti. Taip mažėja tikimybė pateikti klaidingą atsakymą su per dideliu užtikrintumu.

Minimalios atsakytų užklausų dalies sąlyga saugo nuo dirbtinio rezultato pagerinimo. Sistema negali tiesiog nepateikti atsakymo daugumai sunkių atvejų ir dėl to atrodyti geresnė. Ji turi atsakyti bent į 0,8 dalį užklausų, todėl patikimumo slenkstis vertinamas kaip kokybės ir atsakymų kiekio kompromisas.

Atsakymo nepateikimas naudingas tik tada, kai kartu matoma, kokiai užklausių daliai sistema vis dar atsako. Todėl slenkstis buvo parenkamas taip, kad mažintų klaidų riziką, bet neleistų sistemai per dažnai tylėti.

5.4.6. Kelių pavyzdžių režimas ir laipsniškas klasių pridėjimas

Kelių pavyzdžių bandymas tikrino, ar paieška pagal panašumą išlieka naudinga, kai naujos klasės atsiranda palaipsniui. Praktinis klausimas buvo, ar užtenka pridėti kelis naujos klasės pavyzdžius į pavyzdžių rinkinį, nepertreniruojant viso vaizdo modelio. Taip pat tikrinta, ar šiame režime padeda atsakymo nepateikimo taisyklė, kai sistema vis tiek turi atsakyti bent į nustatytą užklausių dalį.

Naudojamas retų klasių atidėjimo bandymas su 10 atsitiktinių sėklų. Kiekvienai sėklai parenkama 10 retų klasių, o jų užklauskos vertinamos atskirai, laikant, kad prieš pridedant naujus pavyzdžius šių klasių pavyzdžių rinkinyje nėra. Po to į pavyzdžių rinkinį pridedama po 1, 3 ir 5 pavyzdžius kiekvienai laikinai atidėtai klasei ir skaičiuojamas rezultatas tik šioms užklauskoms.

5.10 lentelėje pateikti kelių pavyzdžių režimo rezultatai kartu su taisykle, kada sistema neturėtų pateikti atsakymo. Slenkstis buvo parenkamas validacijoje taip, kad makro F1 būtų kuo didesnis, bet sistema vis dar pateiktų atsakymą bent 0,8 visų užklausių daliai. Tada tas pats slenkstis buvo taikomas testavimo aibe.

5.10 lentelė. Kelių pavyzdžių režimo ir laipsniško klasių pridėjimo rezultatų suvestinė

Scenarijus	Variantas	Rezultatai	Ką tai rodo
1 pavyzdys klasei	kNN	makro F1 0,1105±0,0462	Bazinis rezultatas labai žemas.
1 pavyzdys klasei	Su papildomos informacijos perrikiavimu	makro F1 0,2388±0,0519 atkūrimas@5 0,446	Aiškus pagerėjimas net su vienu pavyzdžiu.
3 pavyzdžiai klasei	kNN	makro F1 0,1651±0,0568	Vien kNN išlieka silpnas.
3 pavyzdžiai klasei	Su papildomos informacijos perrikiavimu	makro F1 0,4139±0,1450 atkūrimas@5 0,632	Didelis praktinis šuolis.
5 pavyzdžiai klasei	kNN	makro F1 0,1967±0,0536	Vien kNN vis dar ribotas.
5 pavyzdžiai klasei	Su papildomos informacijos perrikiavimu	makro F1 0,4508±0,1326 atkūrimas@5 0,680	Geriausias atsakymo variantas šiame režime.
Su taisykle, kada atsakymo geriau nepateikti	1 pavyzdys	makro F1 0,2821±0,1043 atkūrimas@5 0,484±0,161 atsakytų atvejų dalis 0,841±0,100 slenkstis 0,013±0,007	Patikimumas dar padidėja.
Su taisykle, kada atsakymo geriau nepateikti	3 pavyzdžiai	makro F1 0,4166±0,1273 atkūrimas@5 0,642±0,127	Nauda išlieka.

		atsakytų atvejų dalis 0,909±0,109 slenkstis 0,007±0,008	
Su taisykle, kada atsakymo geriau nepateikti	5 pavyzdžiai	makro F1 0,4780±0,1420 atkūrimas@5 0,693±0,107 atsakytų atvejų dalis 0,902±0,078 slenkstis 0,008±0,006	Geriausias šio režimo rezultatas.

Kai sistema visada pateikia atsakymą, makro F1 vidurkis ir standartinis nuokrypis per 10 sėklų buvo tokie.

Kai daliai atvejų atsakymas nepateikiamas, slenkstis buvo parenkamas taikant apribojimą, kad sistema turi atsakyti bent į 0,8 dalį užklausų. Žemiau pateikiami vidurkiai ir standartiniai nuokrypiai.

Kai klasės pridamos palaipsniui, papildomų atributų perrikiavimas išlieka naudingas. Net su vienu naujos klasės pavyzdžiu makro F1 buvo didesnis negu vien kNN paieškoje, o pridėjus 3 arba 5 pavyzdžius pagerėjimas išliko. Praktinėje sistemoje tai leidžia pradėti naudoti naują klasę dar neturint daug jos pavyzdžių.

Atsakymo nepateikimo taisyklė veikė ir kelių pavyzdžių režime. Kai sistema turėjo atsakyti bent į nustatytą užklausų dalį, likusiems atsakytams atvejams makro F1 ir atkūrimas@5 buvo didesni. Tai reiškia, kad net naujoms klasėms su keliais pavyzdžiais sistema gali atskirti aiškesnius atvejus nuo tokių, kuriems reikia papildomo tikrinimo.

Šio eksperimento ribojimas yra tai, kad jis sutelktas į retas klases ir todėl turi natūralų rezultatų svyravimą per atsitiktines sėklas.

5.4.7. Mokomas modelis kandidatų tvarkai keisti

Ši patikra tikrino, ar papildomų atributų perrikiavimą verta pakeisti mokomu modeliu. Toks modelis po pirminės paieškos gauna kandidatų sąrašą ir bando iš naujo sudėlioti kandidatų eilę. Lyginta su paprastesnėmis taisyklėmis, kad būtų aišku, ar mokymas duoda ne tik sudėtingesnę, bet ir geresnę sprendimą.

5.11 lentelėje pateikti mokomo perrikiavimo bandymai. Modelis buvo mokomas iš K geriausių kandidatų porų ir jų požymių, tokių kaip kosinusinis panašumas, svorio ir skersmens atstumai, metų atitikimas ir kandidato vieta sąrašė. Palyginimui palikta ta pati paieškos bazė, kad skirtumas būtų siejamas būtent su kandidatų tvarkos keitimu.

5.11 lentelė. Mokomo perrikiavimo bandymų rezultatų suvestinė

Patikra	Variantas	Rezultatai	Ką tai rodo
Paprastas mokomas perrikiuotojas	Taisyklinis perrikiavimas, 100 kandidatų	makro F1 0,7981 atkūrimas@5 0,9782	Atspirties taškas.

Paprastas mokomas perrikuotojas	Logistinė regresija	makro F1 0,7403	Bazės neaplenkė.
Paprastas mokomas perrikuotojas	GBDT	makro F1 0,6952	Bazės neaplenkė.
Porinis verifikatorius	Vienintelis mechanizmas	makro F1 0,4949	Per silpnas.
Porinis verifikatorius	Su sunkiomis neigiamomis	makro F1 0,6023	Pagerėjo, bet vis tiek per žemas.
Porinis verifikatorius	Pakartotinis bandymas	makro F1 0,5412 atkūrimas@1 0,9054	Išliko nestabilus.
Porinis verifikatorius	Kaip papildomas signalas	makro F1 0,7650	Taisyklių neaplenkė.
LightGBM LambdaRank	500 kandidatų sąrašas	makro F1 0,7332 atkūrimas@5 0,9777	Išlaiko aukštą lygį pagal tai, kiek užklausų sistema atsako, bet silpnai kelia teisingą kandidatą į pirmą vietą.
LightGBM LambdaRank	Bazinis taisyklinis variantas	makro F1 0,8190 atkūrimas@5 0,9787	Likęs geriausias sprendimas.

Mokomas modelis kandidatų tvarkai keisti šiame rinkinyje buvo mažiau stabilus negu paprastos taisyklės. Tikėtina priežastis yra per didelis prisitaikymas prie riboto kandidatų sąrašo, ypač kai papildomi laukai nepilni arba triukšmingi. Todėl galutiniame sprendime paliktas paprastesnis perrikiavimas, o mokomas variantas paliktas tolesniems darbams.

Neigiamas rezultatas čia padeda pagrįsti sprendimo pasirinkimą. Sudėtingesnis modelis nėra geresnis vien dėl to, kad jis mokomas iš duomenų. Jei įėjimo laukai triukšmingi, o mokymas vyksta tik su K geriausių kandidatų sąrašu, modelis gali išmokti nestabilius sutapimus. Todėl šiame darbe paliktas perrikiavimas pagal suprantamas taisykles.

5.4.7.1. Neuroninis porinis verifikatorius

Išbandytas ir porinis verifikatorius. Po pirminės vaizdo paieškos kiekviena užklausos ir kandidato pora įvertinama atskiru neuroniniu modeliu. Modelis prognozuoja, ar kandidatas greičiausiai yra teisingas, o pagal šį įvertį keičiama kandidatų eilė.

Verifikatorius mokytas naudojant fiksuotus vaizdinius požymių vektorius iš bazinio koduotojo ir porinius požymius. Naudotas požymių vektorių absoliutus skirtumas ir sandauga, kosinusinis panašumas, kandidato vieta pradiniam sąrašo ir dalis papildomos informacijos požymių, pavyzdžiui, svorio bei skersmens atstumai, metų atitikimas ir trūkstumų žymės. Kiekvienai užklausiai sudarytas pradinis kandidatų sąrašas, o mokymo metu naudoti tiek lengvi, tiek sunkūs neigiami pavyzdžiai. Taip modelis buvo verčiamas skirti subtilius skirtumus tarp labai panašių kandidatų.

Poromis mokomas verifikatorius nepakeitė paprasto perrikiavimo. Tikėtina, kad naudoti poriniai požymiai buvo per mažai detalūs, todėl modelis nematė pakankamai smulkių skirtumų tarp labai panašių monetų. Dėl to verifikatorius paliekamas kaip tolesnė kryptis, o ne kaip pagrindinis sprendimas.

5.4.7.2. Rikiavimo modelis su LightGBM LambdaRank

Toliau tikrinta, ar kandidatų eilę stabiliau keistų rikiavimo modelis, kuris mokosi tvarkyti visą vienos užklausos kandidatų sąrašą, o ne klasifikuoja pavienes poras. Šis bandymas turėjo sumažinti persimokymo riziką, matytą 7a ir 7b bandymuose, ir kartu patikrinti H6 prielaidą.

Kiekvienai užklausiai sudaromas realus kandidatų sąrašas iš pirmo žingsnio paieškos, atrenkant kandidatus pagal vaizdinį panašumą su mokymo pavyzdžių rinkiniu. Mokymo duomenys konstruojami tik iš šio realaus K geriausių kandidatų sąrašo.

Dabartinis požymių rinkinys buvo per siauras ir per daug panašus į bazinę taisyklę. Modelis labiau išmoko nestabilius kompromisus negu naudingą kandidatų eilės pakeitimą. Šis rezultatas rodo, kad vien pridėti mokomą antrą etapą neužtenka. Reikia daugiau informacijos apie kandidatus ir griežtesnės apsaugos nuo persimokymo.

Naudoti požymiai buvo paprasti ir lengvai paaiškinami. Į juos įėjo kosinusinis panašumas, kandidato vieta sąraše, svorio ir skersmens atstumai, metų atitikimas, kai kurie kategorinių laukų sutapimai ir trūkstamumo žymės. Tokia schema išlaikė ryšį su baziniu sprendimu, bet neleido modeliui matyti daug smulkesnių skirtumų.

Mokymui naudotas LightGBM LambdaRank su ankstyvu stabdymu pagal validavimo aibę. Nustatymai buvo ribojami taip, kad modelis liktų mažos rizikos ir veiktų CPU režimu. Buvo ribojamas medžių gylis, lapų skaičius ir mažiausias pavyzdžių skaičius lape. Mokymo metu stebėta NDCG, o galutinis palyginimas atliktas pagal makro F1, atkūrimą@5 ir klasių dažnumo grupes.

Vertinime šis modelis buvo lyginamas su perrikiavimu pagal paprastas taisykles, paliekant tas pačias paieškos ir kandidatų sąrašo sudarymo sąlygas. Buvo žiūrima vieno atsakymo makro F1, atkūrimas@5 ir dažnų, vidutinio dažnio bei retų klasių grupės. Kai reikėjo, papildomai tikrintas ir patikimumo mechanizmas su sąlyga, kad sistema atsako bent į 0,8 dalį atvejų.

Kad modelis nepersimokytų, nustatymai buvo parenkami tik validavimo aibėje, o testavimo rezultatai skaičiuojami vieną kartą su jau užfiksuotais nustatymais. Stebėti ir aiškūs perspėjimo požymiai. Tai didelis pagerėjimas tik validacijoje, sumažėjęs atkūrimas@5 arba ryškiai pablogėję retų klasių rezultatai.

Sudėtingesnis mokomas modelis kandidatų tvarkai keisti šiame darbe bazinio sprendimo nepralenkė. Dėl to galutiniame variante paliktas paprastesnis ir aiškiau paaiškinamas perrikiavimas pagal papildomą informaciją.

5.4.8. Etikečių smulkumo patikra

Paskutinėje šio skyriaus patikroje žiūrėta, ar dalis klaidų kyla ne iš paieškos logikos, o iš per smulkių etikečių. Kitaip tariant, tikrinta, ar rezultatai pagerėtų, jei labai artimos klasės būtų sujungtos į stambesnes grupes.

5.12 lentelėje pateikta etikečių smulkumo patikra. Joje etiketės sujungtos arba pervadintos po eksperimento į stambesnes grupes, pavyzdžiui, sujungiant žinomas klasių variacijas. Tuo metu nekeičiami nei požymių vektoriai, nei perrikiavimo mechanizmas.

5.12 lentelė. Etikečių smulkumo patikros rezultatų suvestinė

Variantas	makro F1	Ką tai rodo
Originalios etiketės	0,8190	Bazinis rezultatas.
Stambesnės grupės po peržymėjimo	0,8397	Dalis klaidų kyla iš per smulkių etikečių.

Šis pagerėjimas rodo, kad dalis klaidų yra ne vien algoritmo problema. Kai kurios klaidos atsiranda todėl, kad sistema pateikia labai artimą kandidatą, kuris praktikoje būtų naudingas, bet pagal dabartinį vertinimą vis tiek laikomas klaida. Todėl vertinant rezultatus svarbu aiškiai pasakyti, kokio detalumo atpažinimas šiame darbe laikomas tikslu.

Šis eksperimentas apriboja galutines išvadas. Jis parodo, kad dalis klaidų kyla ne vien iš algoritmo ribotumų, o iš to, kaip suformuluotas uždavinys ir etiketės. Todėl galutines išvadas reikia sieti ne tik su modeliu, bet ir su užduoties formuluote.

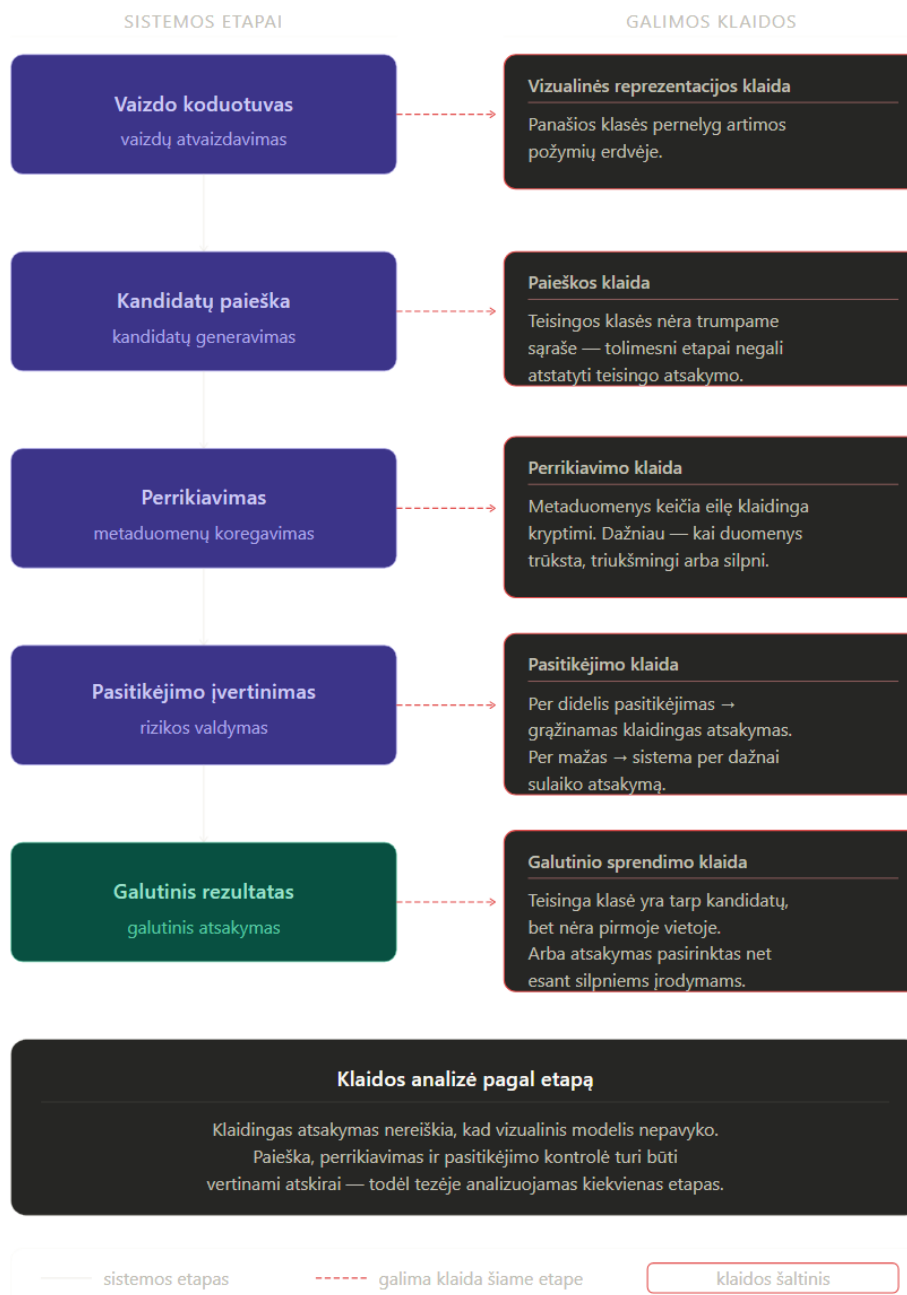
5.4.9. Klaidų analizė

Klaidų analizė susieja skaičius su konkrečiomis sprendimo vietomis. Ji parodo, ar klaida prasideda pradinėje vaizdo paieškoje, ar atsiranda vėliau, kai kandidatų eilė keičiama pagal papildomus atributus.

Klaidos suskirstytos į keturias grupes. Pirma, teisinga klasė išvis nepatenka į kandidatų sąrašą. Antra, ji sąrašė yra, bet lieka per žemai. Trečia, perrikiavimas pablogina teisingo kandidato vietą. Ketvirta, atsakymo nepateikimo taisyklė atmeta per daug arba per mažai atvejų. Kiekvienai grupei paimti konkretūs pavyzdžiai.

Klaidos nagrinėtos dviem būdais. Pirma žiūrėta, kurios klasės dažniausiai painiojamos tarpusavyje. Antra, peržiūrėti konkretūs pavyzdžiai su vaizdais ir kandidatų eile.

5.2 pav. parodyta, kuriose sistemos eigos vietose gali atsirasti klaidos ir kodėl blogas galutinis atsakymas dar nereiškia, kad suklydo tik vaizdo modelis.



5.2 pav. Galimų klaidų vietos sistemos eigoje

Pagal dažniausių supainiojimų suvestinę, didžiausios klaidos kartojasi tarp panašios vertės monetų ir artimų klasių grupių.


Dažniausi supainiojimai buvo tarp labai panašios vertės monetų ar grupių. Pavyzdžiui, DenominationCSco buvo painiojama su DenominationBSCO 7 kartus iš 63, Didrachm su Tetradrachm 6 kartus iš 9, Hemidrachm su Drachma 2 kartus iš 6, Hemiobol su Dichalkon 2 kartus iš 8, o Chalkous su DenominationBSCO 2 kartus iš 5.


Šis rezultatas atitinka didelio netolygumo problemą. Kai klasė turi tik kelis mokymo pavyzdžius, pavyzdžiui, 6-9, keli supainiojimai iš karto smarkiai sumažina tos klasės tikslumą ir F1.


Toliau 5.3 pav. Klaidų analizės pavyzdžiai I ir 5.4 pav. Klaidų analizės pavyzdžiai II pateikti visi septyni analizuoti klaidų atvejai.


Vizualiai labai panašios klasės
Didrachm supainiotas su Tetradrachm, o teisinga klasė liko 2 vietoje.


Užklausa


Tikroji klasė: Didrachm


Klaidingas pirmas kandidatas



1 vieta: Tetradrachm



2 vieta: Didrachm



Teisinga klasė nebuvo pirma

Papildoma informacija stumia klaidingą
Chalkous supainiotas su DenominationCSco, nors teisinga

Užklausa



Tikroji klasė: Chalkous



Klaidingas pirmas kandidatas



1 vieta: DenominationCSco


Trūkstamas skersmuo silpnina signalą
DenominationCSco supainiota su DenominationBSco, o teisinga klasė nukrito į 6 vietą.

Užklausa


Tikroji klasė: DenominationCSco


Klaidingas pirmas kandidatas



1 vieta: DenominationBSco





Teisinga klasė į trumpą kandidatų sąrašą nepateko.

Klaida dar paieškos etape
Obol supainiotas su DenominationBSco, o teisinga klasė į 6 vietą

Užklausa


Tikroji klasė: Obol


Klaidingas pirmas kandidatas



1 vieta: DenominationBSco


5.3 pav. Klaidų analizės pavyzdžiai I


5.3 pav. Klaidų analizės pavyzdžiai Iparodyti keturi atvejai. Tai vizualiai labai panašios klasės, papildomos informacijos nulemtas klaidingas pirmos vietos kandidatas, atvejis be skersmens ir klaida, prasidėjusi dar pirminėje paieškoje.


Artima klasė lieka pirmoje vietoje
Hemidrachm supainiotas su Drachma, o teisinga klasė liko 3 vietoje.


Užklausa



Tikroji klasė: Hemidrachm


Klaidingas pirmas kandidatas


1 vieta: Drachma






3 vieta: Hemidrachm



Teisinga klasė nebuvo pirma

Pirmoje vietoje labai artima grupė
Hemiobol supainiotas su Dichalkon, o teisinga klasė liko ka

Užklausa



Tikroji klasė: Hemiobol



Klaidingas pirmas kandidatas



1 vieta: Dichalkon


Galutinis balsavimas pakeičia atsakymą
DenominationDSco atveju pirmoje vietoje buvo teisinga klasė, bet galutinis atsakymas perėjo į DenominationCSco.


Užklausa



Tikroji klasė: DenominationDSco


Teisinga klasė buvo sąrašo viršuje


1 vieta: DenominationDSco


2 vieta: DenominationCSco


Galutinis atsakymas perėjo į šią grupę



5.4 pav. Klaidų analizės pavyzdžiai II

5.4 pav. Klaidų analizės pavyzdžiai Iparodyti likę trys atvejai. TaiHemidrachm ir Hemiobol supainiojimai, kai teisinga klasė lieka kandidatų sąrašė, ir DenominationDSco atvejis, kuriame artima grupė perima galutinį atsakymą.

1. Panašios vertės monetos yra viena svarbiausių klaidų grupių. Tokiais atvejais teisingas kandidatas dažnai jau yra geriausių kandidatų sąrašė, bet ne pirmoje vietoje. Kandidatų sąrašas naudotojui vis dar būtų naudingas, tačiau vieno atsakymo sprendimas lieka dviprasmiškas.
Pavyzdžiui, Didrachm buvo supainiotas su Tetradrachm, o teisinga klasė liko 2 vietoje, kai balų skirtumas buvo apie 0,022. Hemidrachm buvo supainiotas su Drachma, o teisinga klasė liko 3 vietoje, kai balų skirtumas buvo apie 0,028. DenominationDSc0 buvo supainiotas su DenominationBSc0, o teisinga klasė liko 2 vietoje, kai balų skirtumas buvo apie 0,014. Hemiobol buvo supainiotas su Dichalkon, o teisinga klasė liko 2 vietoje, kai balų skirtumas buvo apie 0,055. Šie atvejai rodo, kad klaida ne visada reiškia visišką nesėkmę. Teisingas kandidatas būna labai arti pirmos vietos, todėl kandidatų sąrašas vis dar turi praktinės naudos. Tačiau vieno atsakymo vertinimas tokią situaciją skaičiuoja kaip klaidą. Būtent čia reikalingas patikimumo žingsnis, nes jis gali parodyti, kad pirmas atsakymas nėra pakankamai atskirtas nuo kitų.
 2. Kai papildomi atributai labai gerai sutampa su neteisingu kandidatu, jie gali iškelti jį į pirmą vietą. Tokios klaidos atsiranda ne todėl, kad vaizdo paieška visiškai prasta, o todėl, kad klasių ribos pagal papildomus laukus realiai persidengia.
Vienas pavyzdys buvo Chalkous, supainiotas su DenominationCSc0. Teisinga klasė liko antroje vietoje. Pirmos vietos kandidatas pagal papildomus atributus atrodė labai tinkamas, nors vaizdų skirtumas nebuvo aiškus. Tai rodo, kad dalyje atvejų papildomi atributai gali perkelti neteisingą kandidatą į pirmą vietą.
Tokiais atvejais papildomi atributai nepakankamai atskiria konkrečias klases. Perrikiavimo taisyklė tada gali per daug sustiprinti neteisingą, bet pagal papildomus laukus panašų kandidatą.
 3. Skersmens trūkumas silpnina artimų grupių atskyrimą. Pogrupių analizė jau parodė, kad testavimo aibėje pavyzdžiai be skersmens yra sunkesni. Konkrečiuose klaidų pavyzdžiuose matyti atvejais, kai skersmens trūkumas sutampa su supainiojimu tarp gretimų grupių.
Kitas pavyzdys parodė, ką reiškia trūkstamas skersmuo. DenominationCSc0 buvo supainiota su DenominationBSc0, o teisinga klasė liko tik šeštoje vietoje. Papildomos informacijos įrašė skersmuo buvo nežinomas, todėl skersmuo šioje vietoje negalėjo padėti atskirti panašių klasių.
Kai trūksta skersmens, papildomi atributai mažiau padeda atskirti kandidatus. Tada sprendimą labiau lemia vaizdo panašumas ir datos. Jei šios reikšmės tarp gretimų grupių panašios, klaidos tikimybė didėja.
 4. Kita klaidų grupė atsiranda tada, kai teisingos klasės išvis nėra kandidatų sąrašė. Tokiu atveju perrikiavimas nebegali padėti, nes jis gali keisti tik tų kandidatų eilę, kurie jau pateko į sąrašą.
Vienas toks pavyzdys buvo Obol, supainiotas su DenominationBSc0. Teisinga klasė nepateko į K geriausių kandidatų sąrašą, o teisingos klasės vieta tarp K geriausių kandidatų nebuvo rasta. Tai reiškia, kad problema čia prasidėjo dar paieškos pakopoje.
- 5.3 pav. Klaidų analizės pavyzdžiai I ir 5.4 pav. Klaidų analizės pavyzdžiai II pateikti visi septyni analizuoti atvejai, o priede palikta išsamesnė jų reitingų informacija.
- Daliai užklausų teisingos klasės pavyzdžiai pagal požymių vektorius lieka per toli. Priežastis gali būti prasta vaizdo kokybė arba labai didelis vizualinis panašumas su kitomis grupėmis. Tokiais atvejais pagrindinė problema yra paieška, o ne papildomų atributų perrikiavimas.

Likusios klaidos telkiasi keliose grupėse. Dalis atsiranda tarp labai artimų klasių, kai teisingas kandidatas yra sąrašė, bet ne pirmoje vietoje. Kita dalis kyla tada, kai teisinga klasė į kandidatų sąrašą išvis nepatenka. Dar viena dalis siejasi su trūkstamais papildomais atributais, ypač skersmeniu.

Kai teisinga klasė nepatenka į kandidatų rinkinį, vien papildomų atributų perrikiavimas problemos neišspręs. Tokia klaida reiškia, kad užklauskos vaizdas pagal požymių vektorius liko per toli nuo teisingos klasės pavyzdžių arba pats vaizdas buvo per triukšmingas. Tolesni patobulinimai tokioje vietoje turėtų gerinti požymių vektorius, didinti kandidatų rinkinį arba pridėti daugiau vaizdinių požymių.

Papildomų atributų nepilnumas, ypač skersmens trūkumas, yra pasikartojanti riba atskiriant artimas grupes. Tai paaiškina, kodėl dalis supainiojimų lieka net ir po perrikiavimo, ir kodėl rezultatus būtina aiškinti kartu su pogrupių analize pagal žinomą ir nežinomą skersmenį.

5.5. Eksperimentinės dalies išvados

Eksperimentai parodė, kad vienas klasifikatorius, kuris iš karto parenka vieną klasę, šiam rinkiniui netiko. Jo makro F1 buvo 0,4258, o vien vaizdo kNN paieškos makro F1 buvo 0,4630. Tačiau vaizdo paieška dažnai turėjo teisingą klasę tarp kelių artimiausių kandidatų, todėl ją buvo verta naudoti kaip pirmą etapą. Kai po šio etapo kandidatų eilė buvo pakeista pagal metus, svorį ir skersmenį, makro F1 pakilo iki 0,8190.

Patikrinti literatūroje siūlomi sudėtingesni keliai šiame rinkinyje nepakeitė geriausio varianto. FastAP davė 0,7335, kontekstinis perrikiavimas 0,7894, o vidutinis užklauskos išplėtimas 0,7903. Visi šie skaičiai buvo mažesni už bazinį variantą su 0,8190. Tai reiškia, kad metodą reikėjo rinktis pagal šio rinkinio klaidas, o ne pagal metodo sudėtingumą.

Perrikiavime daugiausia padėjo metų intervalo sutapimas. Svoris ir skersmuo taip pat padėjo, kai reikšmės buvo žinomos, bet jų trūkumas sumažino naudą. Pogrupių analizė parodė, kad nežinomas skersmuo apsunkina vizualiai artimų klasių atskyrimą, todėl šio duomenų lauko pilnumas tapo viena realių sistemos ribų.

Kandidatų sąrašo dydžio patikra nebuvo tik nustatymo paieška. Ji parodė, kiek sprendimas priklauso nuo to, ar teisinga klasė dar lieka tarp kandidatų po pirmos paieškos. Didinant sąrašą nauda buvo nedidelė. Bazinis 500 kandidatų variantas davė 0,8190, o kandidatų įvairinimas su didesniu pradiniu sąrašu davė 0,8221. Dėl to šis pakeitimas laikomas nedideliu mastelio patobuliniu, o ne pagrindine darbo išvada.

Patikimumo patikra buvo skirta praktinei situacijai, kai neteisingas atsakymas yra blogiau už nepateiktą atsakymą. Todėl buvo vertinama ne tik, ar teisinga klasė yra tarp 5 kandidatų, bet ir kokiais užklauskų daliais sistema vis tiek pateikia atsakymą. Kalibruota versija leido sistemos pasitikėjimą išreikšti tikimybe. Brier balas matavo, kiek ši tikimybė skiriasi nuo realaus teisingumo, o ECE matavo vidutinę tikimybių paklaidą.

Mokomas modelis kandidatų eilei keisti nepasiteisino šiame etape. Jo makro F1 buvo 0,7119, o taisyklinis perrikiavimas davė 0,8190. Dėl to mokomas perrikiuotojas nebuvo atmestas visam laikui, bet šiame darbe jam trūko stabilesnių įvesties duomenų arba pakartotinių mokymo patikrų.

Rezultatai patvirtino, kad šiame rinkinyje verta atskirti du etapus. Pirmame etape randami artimi kandidatai pagal vaizdą. Antrame etape jų eilė keičiama pagal metus, svorį ir skersmenį. Patikimumo taisyklė nėra trečias atpažinimo metodas. Ji tik nusprendžia, kada parengtas atsakymas per rizikingas rodyti kaip galutinį.

Didžiausią pagerėjimą davė ne vienas naujas modelis, o šių etapų sujungimas. Vaizdo paieška išlaikė teisingą klasę kandidatų sąraše, perrikiavimas ją dažnai perkėlė į pirmą vietą, o patikimumo patikra sumažino klaidingai užtikrintų atsakymų riziką. Toks derinys pasiekė 0,8190 makro F1 ir geriau tinka situacijai, kai naudotojui reikia patikimo monetos kandidatų sąrašo, o ne vieno spėjimo bet kokia kaina.

5.6. Kas gali riboti rezultatų patikimumą

Rezultatų patikimumą labiausiai riboja mažos retos klasės, nepilni monetos laukai ir vienas fiksuotas duomenų padalijimas. Dėl šių ribų išvadas reikia vertinti kartu su klaidų analize, o ne vien pagal bendrą rezultatą.

Mažose klasėse kelios klaidos gali smarkiai pakeisti pagal klases skaičiuojamus balus, todėl rezultatai vertinti kartu su dažnų, vidutinių ir retų klasių grupėmis bei klaidų pavyzdžiais. Nepilni laukai, ypač skersmuo, mažina perrikiavimo naudą, todėl atskirai tikrinti atvejai su žinomu ir nežinomu skersmeniu. Dalis klaidų gali kilti ir iš etikečių smulkumo, todėl buvo žiūrima, kas pasikeičia naudojant stambesnes grupes. Nustatymai parinkti validavimo aibėje, atskiroje duomenų dalyje, skirtoje derinimui, kad testas nebūtų priderintas. Kadangi mokymas turi atsitiktinumo, ateityje reikia pakartoti apmokymus su skirtingomis sėklomis.

6. Išvados

1. Vaizdo paieška šiame darbe pasiteisino kaip pirmas, bet ne galutinis atpažinimo etapas. Vien kNN paieškos makro F1 buvo 0,4630, todėl jos pirmo kandidato neužteko laikyti atsakymu. Tačiau pagrindiniame variante teisinga moneta pateko į 5 kandidatų sąrašą 97,87 % atvejų. Tai leido antrajame etape taisyti kandidatų eilę, o po metų, svorio ir skersmens panaudojimo makro F1 pakilo iki 0,8190.
2. Abi monetos pusės turi likti sprendime. Naudojant tik priekinę pusę makro F1 sumažėjo iki 0,7426, o naudojant tik galinę pusę iki 0,7520. Abi pusės kartu davė 0,8190. Tai rodo ne vien skaičių skirtumą, bet ir praktinę priežastį. Vienoje pusėje gali būti valdovo atvaizdas ar stambus simbolis, kitoje tekstas, vertė ar smulkūs ženklai, todėl vienos pusės pasirinkimas praranda dalį atpažinimui reikalingos informacijos.
3. Metai, svoris ir skersmuo geriausiai veikė tada, kai jie keitė kandidatų eilę po vaizdo paieškos, o ne griežtai atmetė kandidatus iš anksto. Didžiausią naudą davė metų laikotarpio sutapimas. Svoris ir skersmuo padėjo tada, kai reikšmės buvo žinomos. Kai skersmens trūko, sistema sunkiau atskyrė labai artimas klases, todėl papildomi laukai turi būti naudojami kartu su jų pilnumo patikra.
4. Sudėtingesni metodai šiame rinkinyje nebuvo geresni vien todėl, kad jie sudėtingesni. FastAP davė 0,7335, kontekstinis perrikiavimas 0,7894, o mokomas kandidatų perrikiuotojas 0,7119. Taisyklinis variantas davė 0,8190, o paprastas kandidatų įvairinimas pakėlė rezultatą tik iki 0,8221. Todėl geriausias pasirinkimas šiame darbe buvo ne didinti modelių sudėtingumą, o išlaikyti vaizdo paiešką ir kontroliuoti, kaip po jos naudojami papildomi monetos laukai.
5. Patikimumo ir kelių pavyzdžių patikros parodė, kad sprendimą galima naudoti atsargiau negu paprastą vieno atsakymo klasifikatorių. Kai pirmi kandidatai yra per arti, sistema gali nepateikti vieno galutinio atsakymo ir taip sumažinti klaidingai užtikrintų atsakymų riziką. Kelių pavyzdžių bandymas parodė, kad naują retą klasę galima įtraukti papildant paieškos rinkinį keliais pavyzdžiais be viso vaizdo modelio mokymo iš naujo. Vis dėlto su vienu pavyzdžiu rezultatas dar lieka silpnas, o su 3 ar 5 pavyzdžiais papildomos informacijos perrikiavimas tampa naudingesnis.
6. Klaidų analizė parodė, kur panašus sprendimas klysta. Jei teisinga klasė nepatenka į kandidatų sąrašą, perrikiavimas jos nebegali gražinti. Jei teisinga klasė yra sąrašė, bet lieka ne pirma, reikia geresnio eilės keitimo pagal papildomus laukus. Jei trūksta skersmens arba klasės skiriasi tik smulkiais detalėmis, reikia geresnių duomenų arba aiškesnio etikečių detalumo. Dėl to klaidos buvo vertinamos pagal sprendimo grandinės vietą, o ne tik kaip bendras neteisingų atsakymų skaičius.

7. Ribojimai ir tolesni darbai

Šio darbo ribojimai susiję su trimis vietomis. Pirma, klasės pasiskirsčiusios labai netolygiai. Kai reta klasė turi mažai pavyzdžių, kelios klaidos gali smarkiai pakeisti pagal klases skaičiuojamus balus, todėl rezultatus reikia žiūrėti atskirai dažnoms, vidutinėms ir retoms klasėms. Antra, papildomi monetos laukai nėra pilni. Skersmens trūkumas mažina galimybę atskirti vizualiai artimas klases, ypač jei trūkstamos reikšmės susitelkia keliose klasėse. Trečia, dalis klasių apibrėžtos labai smulkiai. Kai dvi etiketės skiriasi tik maža detale, reikia nuspręsti, ar tokio smulkumo atpažinimas yra darbo tikslas.

Tolesni darbai turi tikrinti konkrečias sprendimo grandinės vietas. Vien pridėti daugiau duomenų nepakanka, jeigu neaišku, ar tai pagerina vaizdo paiešką, kandidatų eilės keitimą, patikimumo patikrą ar tik bendrą vidurkį.

- 1) Naujų ir retų klasių pridėjimą verta tirti kaip sistemos atnaujinimą. Reikia patikrinti, kas pasikeičia, kai naujai klasei į paieškos rinkinį pridedama po 1, 3 arba 5 pavyzdžius. Toks bandymas parodytų, ar užtenka papildyti paieškos rinkinį, ar vaizdo modelį reikia mokytį iš naujo. Tą patį bandymą reikia kartoti su keliomis atsitiktinėmis sėklomis, kad rezultatas nebūtų vieno padalijimo atsitiktinumas.
- 2) Papildomų laukų nepilnumą reikia tirti kontroliuojamai. Skersmens reikšmių užpildymas būtų prasmingas tik tada, jei būtų palyginti trys atvejai. Viename skersmuo žinomas, kitame jo nėra, trečiame jis atkuriamas iš kitų šaltinių. Tada būtų galima patikrinti, ar mažėja klaidos tarp vizualiai artimų klasių, o ne tik pagerėja bendras vidurkis.
- 3) Patikimumo slenksčius reikia tikrinti atskirai dažnoms, vidutinėms ir retoms klasėms. Vienas bendras slenkstis gali palikti sistemą naudingą dažnoms klasėms, bet per dažnai nepateikti atsakymų retoms klasėms. Todėl tolesniame darbe verta lyginti bendrą slenkstį su atskirais slenksčiais grupėms ir kartu rodyti klaidų riziką bei atsakytų užklausų dalį.
- 4) Etikečių detalumą reikia tikrinti atskirai nuo modelio kokybės. Jei dvi klasės praktiškai skiriasi tik smulkiomis detalėmis, stambesnės grupės arba aiški hierarchija parodytų, ar klaida kyla iš silpnos paieškos, ar iš per smulkiai apibrėžto tikslo.
- 5) Sudėtingesnius modelius verta gražinti tik tada, kai jie tikrinami griežčiau negu šiame darbe. Mokomas perrikuotojas, stipresnis vaizdo koduotuvai ar papildomas verifikatorius turi būti lyginami su taisykliniu variantu per kelis mokymo paleidimus, pagal tas pačias klasių dažnumo grupes ir su ta pačia patikimumo tvarka. Tik tada būtų matyti, ar sudėtingesnis metodas iš tiesų duoda naudą, o ne geriau prisitaiko prie vieno duomenų padalijimo.

8. Literatūros sąrašas

- [1] K. Alexandridis, „Long-tailed Object Recognition - Learning from Imbalanced Image Data,“ King's College London, 2024. [Tinkle]. Available: https://kclpure.kcl.ac.uk/ws/portalfiles/portal/271302902/2024_Alexandridis_Konstantinos_21165577_thesis.pdf. [Kreiptasi 27 01 2026].
- [2] X. Zhao, „Unbiased Shared Feature Learning on Long-tailed Data for Deep-learning based Classification,“ University of Liverpool, 11 2024. [Tinkle]. Available: https://livrepository.liverpool.ac.uk/3187193/1/201532862_Nov2024.pdf. [Kreiptasi 27 01 2026].
- [3] Y. Peng, Y. Zhang ir Y.-m. Cheung, „Semantic-guided Fine-tuning of Foundation Model for Long-tailed Visual Recognition,“ Cornell University, 17 07 2025. [Tinkle]. Available: <https://arxiv.org/abs/2507.12807>. [Kreiptasi 27 01 2026].
- [4] M. Li, Z. Hu, Y. Lu, W. Lan, Y.-m. Cheung ir H. Huang, „Feature Fusion from Head to Tail for Long-Tailed Visual Recognition,“ AAAI Press, 2024. [Tinkle]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/29262>. [Kreiptasi 27 01 2026].
- [5] H. Zhao, Z. Li, W. He ir Y. Zhao, „Hierarchical Convolutional Neural Network with Knowledge Complementation for Long-Tailed Classification,“ Association for Computing Machinery (ACM), 26 04 2024. [Tinkle]. Available: <https://doi.org/10.1145/3653717>. [Kreiptasi 27 01 2026].
- [6] H. Lu, S. Sun, Y. Xie, L. Zhang, X. Yang ir J. Yan, „Rethinking Classifier Re-Training in Long-Tailed Recognition: A Simple Logits Retargeting Approach,“ Cornell University, 01 03 2024. [Tinkle]. Available: <https://arxiv.org/abs/2403.00250>. [Kreiptasi 27 01 2026].
- [7] J. Cui, Z. Zhong, S. Liu, B. Yu ir J. Jia, „Parametric Contrastive Learning,“ Cornell University, 26 07 2021. [Tinkle]. Available: <https://arxiv.org/abs/2107.12028>. [Kreiptasi 27 01 2026].
- [8] F. Cakir, K. He, X. Xia, B. Kulis ir S. Sclaroff, „Deep Metric Learning to Rank,“ IEEE / CVF, 2019. [Tinkle]. Available: https://openaccess.thecvf.com/content_CVPR_2019/papers/Cakir_Deep_Metric_Learning_to_Rank_CVPR_2019_paper.pdf. [Kreiptasi 27 01 2026].
- [9] C. Zhang, G. Almpandis, G. Fan, B. Deng, Y. Zhang, J. Liu, A. Kamel, P. Soda ir J. Gama, „A Systematic Review on Long-Tailed Learning,“ Cornell University, 01 08 2024. [Tinkle]. Available: <https://arxiv.org/abs/2408.00483>. [Kreiptasi 27 01 2026].
- [10] H. Wang, W. Guan, J. Chen, Z. Wang ir D. Zhou, „Towards Heterogeneous Long-tailed Learning: Benchmarking, Metrics, and Toolbox,“ Cornell University, 17 07 2023. [Tinkle]. Available: <https://arxiv.org/abs/2307.08235>. [Kreiptasi 27 01 2026].

- [11] H. Anwar, S. Sabetghadam ir P. Bell, „An Image-Based Class Retrieval System for Roman Republican Coins,“ MDPI, 22 07 2020. [Tinkle]. Available: <https://www.mdpi.com/1099-4300/22/8/799>. [Kreiptasi 27 01 2026].
- [12] D. Swain, V. Rupapara, A. Nour, S. Satapathy, B. Acharya, S. Mishra ir A. Bostani, „A Deep Learning Framework for the Classification of Brazilian Coins,“ Institute of Electrical and Electronics Engineers (IEEE), 2023. [Tinkle]. Available: <https://doi.org/10.1109/access.2023.3321428>. [Kreiptasi 27 01 2026].
- [13] Z. Yang, M. Bastan, X. Zhu, D. Gray ir D. Samaras, „Hierarchical Proxy-Based Loss for Deep Metric Learning,“ IEEE / CVF, 2022. [Tinkle]. Available: https://openaccess.thecvf.com/content/WACV2022/html/Yang_Hierarchical_Proxy-Based_Loss_for_Deep_Metric_Learning_WACV_2022_paper.html. [Kreiptasi 27 01 2026].
- [14] M. Blondel, O. Teboul, Q. Berthet ir J. Djolonga, „Fast Differentiable Sorting and Ranking,“ Cornell University, 20 02 2020. [Tinkle]. Available: <https://arxiv.org/abs/2002.08871>. [Kreiptasi 27 01 2026].
- [15] C. Liao, T. Tsiligkaridis ir B. Kulis, „Supervised Metric Learning to Rank for Retrieval via Contextual Similarity Optimization,“ Cornell University, 04 10 2022. [Tinkle]. Available: <https://arxiv.org/abs/2210.01908>. [Kreiptasi 27 01 2026].
- [16] X. Li, J. Wang, W. Song, Y. Du ir Z. Liu, „Robust Calibrate Proxy Loss for Deep Metric Learning,“ Cornell University, 06 04 2023. [Tinkle]. Available: <https://arxiv.org/abs/2304.09162>. [Kreiptasi 27 01 2026].
- [17] J. Johnson, M. Douze ir H. Jégou, „Billion-scale similarity search with GPUs,“ Cornell University, 28 02 2017. [Tinkle]. Available: <https://arxiv.org/abs/1702.08734>. [Kreiptasi 27 01 2026].
- [18] M. Blachnik ir P. Ciepliński, „Meta-Instance Selection. Instance Selection as a Classification Problem with Meta-Features,“ Cornell University, 20 01 2025. [Tinkle]. Available: <https://arxiv.org/abs/2501.11526>. [Kreiptasi 27 01 2026].
- [19] S. Zambanini, A. Kavelar ir M. Kampel, „Improving Ancient Roman Coin Classification by Fusing Exemplar-Based Classification and Legend Recognition,“ Springer Berlin Heidelberg, 2013. [Tinkle]. Available: https://doi.org/10.1007/978-3-642-41190-8_17. [Kreiptasi 27 01 2026].
- [20] F. Zhu, „Reranking Multi-Modal Retrievers for Knowledge-Based Visual Question Answering,“ University of Cambridge, 08 2024. [Tinkle]. Available: https://www.mlmi.eng.cam.ac.uk/files/2023-2024/zhu_re-ranking_2024_reduced.pdf. [Kreiptasi 27 01 2026].
- [21] Z. Guo, O. Arandjelović, D. Reid, Y. Lei ir J. Büttner, „A Siamese Transformer Network for Zero-Shot Ancient Coin Classification,“ MDPI AG, 25 05 2023. [Tinkle]. Available: <https://www.mdpi.com/2313-433X/9/6/107>. [Kreiptasi 27 01 2026].

- [22] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye ir T.-Y. Liu, „LightGBM: A Highly Efficient Gradient Boosting Decision Tree,“ Neural Information Processing Systems Foundation, 12 2017. [Tinkle]. Available: https://papers.nips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html. [Kreiptasi 12 04 2026].
- [23] F. Zhao, C. Zhang ir B. Geng, „Deep Multimodal Data Fusion,“ Association for Computing Machinery (ACM), 24 04 2024. [Tinkle]. Available: <https://doi.org/10.1145/3649447>. [Kreiptasi 27 01 2026].
- [24] Q. Zhang, Y. Wei, Z. Han, H. Fu, X. Peng, C. Deng, Q. Hu, C. Xu, J. Wen, D. Hu ir C. Zhang, „Multimodal Fusion on Low-quality Data: A Comprehensive Survey,“ Cornell University, 27 04 2024. [Tinkle]. Available: <https://arxiv.org/abs/2404.18947>. [Kreiptasi 27 01 2026].
- [25] S. Li ir H. Tang, „Multimodal Alignment and Fusion: A Survey,“ Cornell University, 26 11 2024. [Tinkle]. Available: <https://arxiv.org/abs/2411.17040>. [Kreiptasi 27 01 2026].
- [26] H. Anwar, S. Anwar, S. Zambanini ir F. Porikli, „Deep Ancient Roman Republican Coin Classification via Feature Fusion and Attention,“ Cornell University, 26 08 2019. [Tinkle]. Available: <https://arxiv.org/abs/1908.09428>. [Kreiptasi 27 01 2026].
- [27] Y. Geifman ir R. El-Yaniv, „SelectiveNet: A Deep Neural Network with an Integrated Reject Option,“ Cornell University, 26 01 2019. [Tinkle]. Available: <https://arxiv.org/abs/1901.09192>. [Kreiptasi 27 01 2026].
- [28] Y. Wang, W. Huang, Y. Wen, S. Liu ir C. Yuan, „Towards Robust Uncertainty Calibration for Composed Image Retrieval,“ OpenReview, 2025. [Tinkle]. Available: <https://openreview.net/forum?id=LSS6JMVgha>. [Kreiptasi 27 01 2026].
- [29] J. Wu, J. Pang ir Q. Huang, „Uncertainty-aware Long-tailed Weights Model the Utility of Pseudo-labels for Semi-supervised Learning,“ Cornell University, 13 03 2025. [Tinkle]. Available: <https://arxiv.org/abs/2503.09974>. [Kreiptasi 27 01 2026].
- [30] „Nomisma datasets,“ Nomisma.org, 2026. [Tinkle]. Available: <http://nomisma.org/datasets>. [Kreiptasi 12 04 2026].
- [31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov ir L.-C. Chen, „MobileNetV2: Inverted Residuals and Linear Bottlenecks,“ IEEE / CVF, 18 06 2018. [Tinkle]. Available: https://openaccess.thecvf.com/content_cvpr_2018/html/Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.html. [Kreiptasi 11 04 2026].
- [32] A. Howard, M. Sanders, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le ir H. Adam, „Searching for MobileNetV3,“ IEEE / CVF, 27 10 2019. [Tinkle]. Available: https://openaccess.thecvf.com/content_ICCV_2019/html/Howard_Searching_for_Mobile_NetV3_ICCV_2019_paper.html. [Kreiptasi 11 04 2026].

- [33] M. Tan ir Q. V. Le, „EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,“ PMLR, 24 05 2019. [Tinkle]. Available: <https://proceedings.mlr.press/v97/tan19a.html>. [Kreiptasi 11 04 2026].
- [34] M. Tan ir Q. V. Le, „EfficientNetV2: Smaller Models and Faster Training,“ PMLR, 18 07 2021. [Tinkle]. Available: <https://proceedings.mlr.press/v139/tan21a.html>. [Kreiptasi 11 04 2026].
- [35] K. He, X. Zhang, S. Ren ir J. Sun, „Identity Mappings in Deep Residual Networks,“ Cornell University, 16 03 2016. [Tinkle]. Available: <https://arxiv.org/abs/1603.05027>. [Kreiptasi 11 04 2026].

Priedai

1 priedas. Bandomųjų variantų ir skaičiavimų santraukos

Šiame priede pateikiami pagrindiniai bandomieji tyrimai ir sprendimai. Kiekviename poskyryje trumpai pasakoma, kas buvo keičiama, kokie nustatymai naudoti ir kokį sprendimą nulėmė gautas rezultatas.

1.1. Bazinis nustatymų rinkinys

Šis bazinis variantas reikalingas kaip atskaitos taškas. Su juo lyginami literatūroje minimi bandymai, kad kiekvienas pakeitimas būtų vertinamas prieš tą pačią stabilią bazę.

Bazinis nustatymas buvo toks. Požymių vektorių paieška vykdoma artimiausių kaimynų metodu, kai $k = 1$ ir kandidatų sąrašo dydis yra 500. Po to kandidatai perrikuojami naudojant papildomą informaciją apie svorį, skersmenį ir metų laikotarpį. Papildomai skaičiuojamas atkūrimas@K, tai yra ar teisinga klasė patenka į pirmas K pozicijų, kai $K \in \{1,3,5,10\}$. Taip pat skaičiuojama rizikos santrauka ir tai, kiek užklausų sistema atsako, apibendrinant rodikliu AURC, o sėkmė apibrėžiama kaip atkūrimas@5.

Testavimo aibėje klasifikatorius, tai yra softmax vieno atsakymo variantas, pasiekė makro F1 0,4258. Prototipai pasiekė makro F1 0,4556. kNN pasiekė makro F1 0,4630. kNN su papildomos informacijos perrikiavimu pasiekė makro F1 0,8190 ir atkūrimą@5 0,9787. Rizikos ir atsakytų atvejų dalies pavyzdys parodė, kad balų skirtumas tarp pirmos ir antros vietos davė AURC 0,0278, o kalibruota tikimybė davė AURC 0,0143.

Papildoma teisingo kandidato patekimo į sąrašą patikra parodė, kad net su 200 kandidatų teisinga klasė beveik visada lieka sąrašė (patekimo dalis $\approx 0,993$). Su 500 kandidatų ši dalis padidėja tik nedaug ($\approx 0,996$). Tai reiškia, kad didesnis kandidatų skaičius dažniau padeda pakeisti teisingo kandidato vietą sąrašė, o ne pirmą kartą jį surasti.

Šis nustatymų rinkinys laikomas baziniu. Visi tolesni bandomieji variantai vertinami pagal tai, ar jie šiuos testavimo rezultatus pagerina, nekeičia ar pablogina.

1.2. Požymių vektorių mokymo pagal panašumą bandomieji variantai

Šioje grupėje tikrinta, ar kitoks požymių vektorių mokymas pagerina vaizdo paiešką taip, kad po metų, svorio ir skersmens panaudojimo pagerėtų galutinis vieno atsakymo rezultatas.

FastAP po papildomos informacijos perrikiavimo davė testavimo makro F1 0,7335, tai yra blogiau nei bazė 0,8190. Todėl ši kryptis šiame darbe neįtraukta kaip pagrindinis gerinimas.

ProxyNCA, tai yra supaprastintas atstovų nuostolis, po papildomos informacijos perrikiavimo pasiekė testavimo makro F1 0,7572. Tai buvo blogiau nei bazinis variantas, todėl ši kryptis šiame darbe neplėtota.

PaCo tipo bandomajame variante, tai yra prižiūrimame mokyme pagal panašias ir skirtingas poras su klasių centrais arba atstovais, po papildomos informacijos perrikiavimo gautas testavimo makro F1 0,7656. Tai blogiau nei bazė, todėl ši kryptis šiame duomenų variante laikoma nepasiteisususia.

SupCon+CE, tai yra prižiūrimo mokymo pagal poras ir klasifikacijos mišinys, pagerino kNN be papildomos informacijos iki testavimo makro F1 0,5143. Po papildomos informacijos perrikiavimo galutinis vieno atsakymo rezultatas buvo 0,7632, todėl liko žemesnis už bazinį. Dėl to ši kryptis palikta kaip sudėtingesnė tolesnių darbų tema.

1.3. Paprastos paieškos taisyklės

Šioje grupėje tikrintos greitos paieškos korekcijos, dažnai minimos literatūroje. Buvo žiūrima, ar jos padeda, ar sugadina smulkių ir vizualiai panašių monetų atskyrimą.

AQE tipo užklauso išplėtimas po papildomos informacijos perrikiavimo davė testavimo makro F1 0,7903 ir atkūrimą@5 0,9750. Tai reiškia, kad vieno atsakymo rezultatas pablogėjo, todėl šis variantas nelaikomas universaliu pagerinimu.

Kontekstinis perrikiavimas pagal kaimynų persidengimą, kai $K = 20$ ir $\lambda = 0,1$, po papildomos informacijos perrikiavimo davė testavimo makro F1 0,7894 ir atkūrimą@5 0,9761. Dėl to jis nebuvo paliktas pagrindinėje sistemos grandinėje.

1.4. Kandidatų sąrašo įvairovės didinimas

Šioje grupėje tikrinta, ar galima sumažinti dažnų klasių kartojimąsi kandidatų sąrašė. Tam pirmiausia imamas didesnis kaimynų sąrašas, o galutiniame sąrašė ribojama, kiek kandidatų galima paimti iš vienos klasės.

Nustatymas – pirmiausia skaičiuotas didesnis pradinis kaimynų sąrašas (2 000), po to suformuotas kandidatų sąrašas 500, ribojant vienos klasės pasikartojimą. Vienos klasės limitas validacijoje parinktas 3.

Testavimo aibėje gauta makro F1 0,8221, kai bazė duoda 0,8190, o atkūrimas@1 0,9671. Šis variantas laikomas realistiška ir mažos rizikos paieškos pakopos plėtra, nes pagerėjimas gautas nekeičiant mokymo.

Papildoma patikra parodė, kad sumažinus pradinį kaimynų sąrašą iki 1 000 įvairinimo nauda išnyko, o testavimo makro F1 sutapo su baze 0,8190. Vadinasi, įvairinimui reikia pakankamai didelio pradinio sąrašo. Tik tada yra iš ko sudaryti įvairesnį galutinį kandidatų sąrašą.

1.5. Paieškos rinkinio retinimas

Šioje grupėje tikrinta, ar paieškos rinkinį galima sumažinti ir taip pagreitinti paiešką neprarandant per daug kokybės.

Nustatymas – paieškos rinkinys sumažinta iki apie 50 % (kiekvienai klasei paliekant arčiausiai klasės vidurkio esančius pavyzdžius), prieš tai 5 634, po retinimo 2 820 pavyzdžių.

Po papildomos informacijos perrikiavimo gautas testavimo makro F1 0,6868 ir atkūrimas@5 0,9437. Tai ženkliai blogiau, todėl šioje duomenų versijoje retinimo netaikyta.

1.6. Dviejų pusių patikra

Šioje grupėje lyginta, kas nutinka naudojant abi monetos puses, tik priekinę pusę arba tik galinę pusę. Taip tikrinta, ar abi pusės duoda papildančią informaciją, ar vienos pusės pakanka.

Naudojant tik priekinę pusę po papildomos informacijos perrikiavimo gautas testavimo makro F1 0,7426.

Naudojant tik galinę pusę po papildomos informacijos perrikiavimo gautas testavimo makro F1 0,7520.

Sprendimas yra aiškus. Abi pusės būtinos smulkaus detalumo identifikavimui, o vienos pusės naudojimas blogina vieno atsakymo kokybę.

1.7. Mokomas modelis kandidatų tvarkai keisti su HGB

Šioje grupėje tikrinta, ar atskirai apmokytas modelis gali geriau pakeisti kandidatų eilę negu rankinė taisyklė. Modelis naudojo vaizdo panašumą ir papildomus monetos laukus, bet buvo lyginamas su paprastesniu taisykliniu perrikiavimu.

Toks mokomas modelis pasiekė testavimo makro F1 0,7119, kai bazė duoda 0,8190. Todėl pagrindinėje sistemoje paliktas taisyklinis perrikiavimas, o šis variantas laikomas tolesnių darbų kryptimi, jei ateityje būtų taikomas griežtesnis mokymo režimas.

Buvo atlikti papildomi bandomieji variantai su neuroniniu poriniu verifikatoriumi. Kai jis naudotas kaip vienintelis kandidatų sąrašo tikslinimo būdas, gauta prasta kokybė. Esant 200 kandidatų sąrašui testavimo makro F1 buvo 0,4949, o su sunkesne neigiamų pavyzdžių atranka 0,6023. Kai verifikatorius naudotas tik kaip papildomas balas ir jungtas su baziniu balu, rezultatas pakilo iki 0,7650, bet bazės 0,8190 vis tiek nepasiekė. Papildomai tikrintas ir variantas, kai verifikatorius tiesiogiai perrikiuoja kandidatų sąrašą. Ten su 200 kandidatų sąrašu gautas 0,5412 makro F1 ir 0,9054 atkūrimas@1. Dėl to ši kryptis šiame darbe laikoma per silpna kaip bazinės sistemos pakaitalas.

1.8. Kiti perrikiavimo ir statistinių korekcijų variantai

Patikrintos trys alternatyvios kNN balsavimo taisyklės. Tai buvo daugumos balsas, svėrimas pagal atstumą ir svėrimas pagal panašumo balą. Šiame duomenų variante geriausiai veikė svėrimas pagal balą, o kiti du variantai mažino vieno atsakymo kokybę. Taip pat patikrinta, ar rezultatą keičia panašumo skalė, kai temperatūra lygi 0,3, 0,5, 1,0 ir 2,0. Ši patikra rezultato nepakeitė, todėl paliktas bazinis nustatymas.

RRF variantas, kuriame kandidatai jungiami pagal vietas sąrašuose, o ne pagal balų sumą, po papildomos informacijos perrikiavimo davė testavimo makro F1 0,6479. Tai buvo gerokai mažiau už bazinį 0,8190 rezultatą, todėl šis variantas nebuvo paliktas.

Netolygaus pasiskirstymo korekcija kNN balsavime, kai naudotas `inv_sqrt_freq`, po papildomos informacijos perrikiavimo davė testavimo makro F1 0,7993. Tai negerino bazės, todėl variantas nebuvo paliktas kaip pagerinimas.

Kai įtrauktas kaldinimo vietos laukas w_{mint} ir naudotas duomenų variantas su šiuo lauku, gautas testavimo makro F1 0,7735. Šioje duomenų versijoje kaldinimo vietos laukas nedavė stabilios naudos, todėl pagrindiniame nustatymų rinkinyje jis nenaudojamas.

Buvo tikrintas griežtas atmetimas pagal papildomą informaciją, pavyzdžiui, pagal leistiną svorio ir skersmens skirtumą, metų intervalų persidengimo reikalavimą ar autoriteto sutapimą. Praktikoje toks atmetimas pasirodė nesaugus, nes didina riziką pašalinti teisingą kandidatą dėl papildomos informacijos netikslumų ar nepilnumo. Dėl to šiame darbe papildoma informacija palikta tik kaip pagalba kandidatų tvarkai pakeisti, o ne kaip atmetimo taisyklė.

Buvo tikrinti keli balo perskaičiavimai pagal papildomos informacijos pasiskirstymą. Vienas variantas pridėjo balą pagal tai, ar kandidato papildoma informacija būdinga jo klasei. Kitas variantas įtraukė vieno atsakymo klasifikatoriaus softmax balą. Trečias variantas suvienodino dalį trūkstamų reikšmių. Stabilaus pagerėjimo negauta. Tikėtina priežastis yra papildomos informacijos netikslumų ir netolygių dažnių sąveika, kuri per stipriai sustiprina dažnas klases. Dėl to šios korekcijos šiame darbe paliktos išjungtos.

1.9. Atsparumo papildomos informacijos trūkumui patikra

Šioje grupėje tikrinta, kaip keičiasi perrikiavimas ir atsakymo nepateikimo slenkstis, kai validacijos aibėje dirbtinai padidinamas trūkstamų metų, svorio ar skersmens laukų kiekis.

Kai validacijoje dirbtinai padidintas trūkstamumas, rezultatas nuosekliai blogėjo. Kai p buvo 0,1, makro F1 siekė 0,7165, kai p buvo 0,2 siekė 0,7129, o kai p buvo 0,3 siekė 0,7021. Baziniame variante makro F1 buvo 0,7358. Tai rodo, kad papildomos informacijos trūkumas mažina kokybę, todėl sistemoje būtina aiškiai aprašyti šį ribojimą ir vertinti pogrupius atskirai.

1.10. Kelių pavyzdžių ir laipsniškas scenarijus

1.1 lentelėje pateikiami kelių pavyzdžių scenarijaus rezultatai. Šio bandymo tikslas buvo įvertinti, ar sistema gali prisitaikyti prie naujų retų klasių papildant paieškos rinkinį keliais pavyzdžiais be pertreniruotės.

1.1 lentelė. Kelių pavyzdžių scenarijaus rezultatų suvestinė

Naujai klasei pridėtų pavyzdžių skaičius	kNN makro F1	kNN su papildomos informacijos perrikiavimu makro F1
1	0,1105±0,0462	0,2388±0,0519
3	0,1651±0,0568	0,4139±0,1450
5	0,1967±0,0536	0,4508±0,1326

Lentelė rodo, kad metų, svorio ir skersmens panaudojimas po paieškos pagerina rezultatą visuose trijuose bandymuose. Pagerėjimas didesnis tada, kai naujai klasei turima daugiau nei vienas pavyzdys.

1.11. Patikimumo balo pavertimas į tikimybę

Šioje grupėje balų skirtumas tarp 1 ir 2 vietos paverčiamas tikimybe, kad atsakymas teisingas. Pagal šią tikimybę tikrinama, kada sistema turėtų nepateikti vieno galutinio atsakymo.

Patikimumo balas taip pat buvo perskaičiuotas į tikimybę, pritaikius S formos kreivę validacijos duomenims. Testavimo aibėje vidutinė kalibravimo paklaida ECE buvo 0,0021. Parinkus 0,95 tikimybės slenkstį, testavimo aibėje gauta, kad sistema atsako į 0,8820 dalį užklausų, o atkūrimas@5 siekia 0,9813. Toks perskaičiavimas leidžia patikimumą aiškinti lengviau, nes slenkstis jau tampa suprantama tikimybe, bet pačios paieškos ir perrikiavimo logikos nekeičia.

Taikant tikimybės slenkstį 0,95, testavimo aibėje tai, kiek užklausų sistema atsako, pagal dažnumo grupes buvo tokia. Dažnoms klasėms gauta 0,8919, vidutinėms 0,7843, o retoms 0,7826. Tai rodo, kad tikimybinis slenkstis labiau sumažina tai, kiek užklausų sistema atsako vidutinėms ir retoms klasėms, todėl šį patikimumo variantą reikia aiškinti kartu su dažnumo grupėmis.

1.12. Dviejų pusių požymių vektorių sujungimo svoris α

Šioje grupėje tikrinta, ar verta fiksuotu svoriu labiau pasitikėti viena monetos puse sujungiant priekinės ir galinės pusės požymių vektorius.

Po papildomos informacijos perrikiavimo gauta taip. Kai α buvo 0,2, testavimo aibėje makro F1 siekė 0,7806. Kai α buvo 0,5, gauta 0,8087. Kai α buvo 0,8, gauta 0,7974. Bazinis rezultatas yra 0,8190, todėl šio tipo fiksuotas svoriavimas bazės negerina ir šiame darbe nenaudojamas.

1.13. Bandymų sugrupavimas pagal temas

1.2 lentelėje bandymai sugrupuoti pagal temas, kad būtų matyti, kokius klausimus apima priedas.

1.2 lentelė. Bandymų sugrupavimo pagal temas suvestinė

Tema	Kiek bandymų	Kas tikrinta	Ką ši grupė padeda suprasti
Kelių pavyzdžių režimas (laipsniškas scenarijus)	20	Prie naujos retos klasės pridėti 1, 3 arba 5 pavyzdžiai, modelio neapmokant iš naujo	Ar sistemą galima greitai papildyti nauja klase, kai turime tik kelis jos pavyzdžius
Mokomas perrikiavimas	17	Po pradinės paieškos kandidatų tvarką nustato atskirai apmokytas modelis, o ne rankinė taisyklė	Ar sudėtingesnis antras etapas iš tikro duoda stabilesnį pagerėjimą
Papildoma informacija	17	Tikrinta, kaip svoris, skersmuo ir metų laikotarpis koreguoja kandidatų tvarką	Kiek papildoma informacija padeda, kai vien vaizdo neužtenka
Papildomas laukas: kaldinimo vieta	14	Įjungtas ir skirtingai pasvertas kaldinimo vietos laukas	Ar kaldinimo vieta verta palikti, kai šis laukas nepilnas

Klasifikatorius (vienas atsakymas)	11	Sistema iškart grąžina vieną atsakymą, be kelių variantų sąrašo	Kiek prarandama, jei atsisakoma kandidatų sąrašo
kNN balsavimo taisyklės	10	Keistas būdas, kaip iš panašiausių pavyzdžių gaunamas galutinis atsakymas	Kuri balsavimo taisyklė duoda geriausią ir stabiliausią rezultatą
Patikimumas (kada sistema neturėtų pateikti atsakymo)	10	Keisti slenksčiai, kada sistema turi geriau atsakymo nepateikti	Kaip keičiasi kompromisas tarp mažesnės klaidų rizikos ir to, kiek užklausų sistema atsako
Vaizdo koduotojo palyginimas	10	Keistas tik požymių vektorių modelis, visa kita paliekant tą patį	Kiek galutinis rezultatas priklauso nuo pasirinkto vaizdo modelio
Paieškos rinkinio mažinimas	7	Mažintas paieškos rinkinys skirtingomis taisyklėmis	Ar sistemą galima pagreitinti stipriai neprarandant kokybės
Dviejų pusių informacija	5	Lyginta tik priekinė pusė, tik galinė pusė ir abi pusės kartu	Ar abiejų monetos pusių tikrai reikia geram atskirumui
Etikečių stambinimas	5	Labai artimos klasės laikinai sujungtos į stambesnes	Ar klaidos kyla iš modelio, ar iš per smulkaus klasių skaidymo
Vektorinių įterpinių mokymo bandymai	5	Keistas požymių vektorių mokymo būdas ir tikslas	Ar kitu mokymo būdu galima pagerinti galutinį rezultatą
Paprastos paieškos taisyklės	4	Taikyti paprasti paieškos patobulinimai, pvz. užklauso išplėtimas, kai užklauso vektorius papildomas artimiausių kandidatų vidurkiu	Ar greitos paieškos korekcijos padeda, ar sugadina artimų klasių atskirtį
Bazinis atskaitos taškas	3	Apmokytas pradinis modelis ir išsaugotas kontrolinis variantas	Su kuo turi būti sąžiningai lyginami visi vėlesni bandymai
Kandidatų sąrašo dydis	3	Keistas kandidatų skaičius, perduodamas į antrą etapą	Koks kandidatų kiekis duoda geriausią balansą tarp kokybės ir kainos
Griežtas atmetimas	3	Kandidatai atmetami anksti, jei papildoma informacija nesutampa	Ar ankstyvas atmetimas saugus, kai laukai nepilni ar netikslūs
Pagrindinis sprendimo variantas	3	Tikrintas pilnas galutinis nustatymų rinkinys su artimais pakeitimais	Ar pagrindinė versija išlieka stabili keičiant artimus parametrus
Papildomos informacijos nepilnumas	3	Dirbtinai padidintas trūkstančių laukų kiekis validacijoje	Kiek sistema jautri situacijoms, kai dalis papildomos informacijos neužpildyta

Patikimumas (kalibravimas)	3	Patikimumo balas paverstas į tikimybę	Ar slenkstį galima nusistatyti aiškiau ir suprantamiau
Neaiškūs atvejai	2	Kai pirmi variantai labai artimi, didintas kandidatų skaičius	Ar tokiose situacijose verta plėsti sąrašą, kad sumažėtų klaidų
Neaiškių atvejų sprendimas	2	Tikrintos ribos, nuo kada įsijungia papildoma kandidatų paieška	Kur praktiškai geriausia riba tarp papildomos naudos ir įnešamo triukšmo
Hierarchinis sprendimas	2	Pirma parenkama stambi grupė, po to tiksli klasė	Ar dviejų žingsnių sprendimas sumažina painiavą tarp panašių klasių
Paieška pagal vaizdą	2	Naudota tik vaizdo panašumo paieška be papildomos informacijos	Kiek pasiekama vien iš vaizdo ir kiek reikia papildomos informacijos
Paieškos kandidatų įvairovė	2	Riboti tos pačios klasės pasikartojimai kandidatų sąrašė	Ar taip pavyksta sumažinti dažnų klasių perteklių sąrašo viršuje
Bazinis palyginimas	1	Išbandytas paprastas prototipų metodas (klasės vidurkis)	Ar toks paprastas metodas gali būti toks pat geras kaip paieška pagal panašiausius pavyzdžius
Mokymo nustatymų patikra	1	Įjungtas vaizdų keitimas mokymo metu	Ar tai duoda realių pagerėjimą šiame duomenų variante
Pograpių analizė	1	Rezultatai atskirti į grupes su žinomu ir nežinomu skersmeniu	Kur sistema stipri ir kur ją labiausiai riboja duomenų trūkumas
Reitingavimo alternatyvos	1	Kandidatų tvarka sudaryta pagal sąrašų vietų sujungimą, ne pagal balų sumą	Ar kita tvarkos sudarymo taisyklė gali pakeisti bazinį sprendimą
Statistikos pagrindu perskaičiuoti balai	1	Taikyti klasių tikėtinumo balo perskaičiavimai pagal papildomą informaciją	Ar toks perskaičiavimas padeda, ar per daug pakelia dažnas klases

Ši lentelė parodo, kokios bandymų kryptys tikrintos priede ir kokį klausimą atsako kiekviena kryptis. Ji padeda greitai susieti atskirus eksperimentų blokus.

1.14. Svarbiausių bandymų grupių paaiškinimai

Šis poskyris paaiškina didžiausias bandymų grupes. 1.2 lentelėje pateikti skaičiai, o čia pasakoma, kas buvo keičiama, kodėl tai tikrinta ir kokia išvada gauta.

Šiuose bandymuose po vaizdo paieškos buvo keičiama, kaip kandidatų eilę koreguoja svoris, skersmuo ir metų laikotarpis. Tikrinta, ar šie laukai padeda atskirti vizualiai panašias monetas, kai

vien vaizdo nepakanka. Jie dažniausiai pagerina rezultatą, bet nauda priklauso nuo to, ar laukai užpildyti ir patikimi.

Požymių vektorių mokymo bandymai. Čia keistas vaizdo modelio mokymo tikslas ir mokymo būdas, kad panašūs pavyzdžiai dar paieškos etape būtų arčiau vieni kitų. Vertinta ne tik tarpinė paieška, bet ir galutinis rezultatas po metų, svorio ir skersmens panaudojimo. Stabilaus pagerėjimo prieš bazinį variantą negauta, todėl ši kryptis nepalikta kaip pagrindinė.

Vaizdo koduotojo palyginimas. Šioje grupėje paliktos tos pačios paieškos ir perrikiavimo taisyklės, o keistas tik vaizdo koduotojas. Taip atskirta, kiek galutinis rezultatas priklauso būtent nuo vaizdo modelio. Išvada svarbi metodikai, nes parodo, kad architektūrų skirtumai yra realūs, bet juos būtina vertinti ta pačia tvarka ir tomis pačiomis sąlygomis.

Paprastos paieškos taisyklės. Tikrinti keli greiti paieškos patobulinimai, kurie teoriškai galėtų pagerinti rezultatą be papildomo mokymo. Žiūrėta, ar jie pagerina pirmą atsakymą ir ar neištrina ribų tarp labai artimų klasių. Šiame darbe tokie patobulinimai dažniau pablogino stabilumą, todėl bazinėje grandyje nebuvo taikomi.

kNN balsavimo taisyklės. Čia keistas tik galutinio balsų sujungimo principas iš artimiausių pavyzdžių. Palyginti variantai su vienodais balsais, su didesniu svoriu artimesniems pavyzdžiams ir su svoriu pagal balą. Praktinė išvada buvo pasirinkti tą taisyklę, kuri duoda ne vienkartinį piką, o nuosekliai gerą rezultatą per kelis paleidimus.

Kandidatų sąrašo dydis. Šioje grupėje lyginta, kiek kandidatų verta perduoti į antrą etapą. Mažas sąrašas spartina skaičiavimą, bet gali neišlaikyti teisingo varianto, o per didelis sąrašas didina triukšmą ir kainą. Todėl čia buvo ieškoma praktinio balanso tarp kokybės ir skaičiavimo sąnaudų.

Paieškos kandidatų įvairovė. Čia tikrinta, ar naudinga riboti pasikartojančius tos pačios dažnos klasės kandidatus sąrašė. Tokia taisyklė palieka daugiau skirtingų alternatyvų ir mažina dažnų klasių perteklių sąrašo viršuje. Ji reikalinga tada, kai duomenyse klasės pasiskirsčiusios netolygiai.

Dviejų pusių informacija. Čia lyginta, kas nutinka naudojant tik priekinę pusę, tik galinę pusę ir abi puses kartu. Papildomai tikrinta, ar verta fiksuotai labiau pasitikėti viena puse. Rezultatas parodė, kad abi pusės duoda papildančią informaciją, o vienos pusės per didelis svoris nėra patikimas sprendimas.

Paieškos rinkinio mažinimas. Ši grupė tikrino, ar galima sumažinti paieškos rinkinio dydį ir taip pagreitinti sistemą. Vertinta, kiek mažinimas kainuoja kokybės požiūriu. Rezultatai parodė, kad per agresyvus mažinimas išmeta reikalingą įvairovę ir blogina galutinį atsakymą.

Papildomas laukas – kaldinimo vieta. Tikrinta, ar kaldinimo vieta pagerina artimų kandidatų atskyrimą. Vertinant paaiškėjo, kad šis laukas dažnai būna nepilnas arba netolygiai sužymėtas, todėl jo poveikis nestabilus. Dėl to ši informacija naudota tik kaip papildoma, bet ne kaip pagrindinė sprendimo atrama.

Griežtas atmetimas. Šiuose bandymuose kandidatai buvo anksti atmetami, jei papildoma informacija nesutapo su užklausa. Nors toks kelias atrodo paprastas ir greitas, jis pavojingas, kai laukai turi klaidų ar trūkumų. Gauta išvada, kad ankstyvas griežtas atmetimas šiame darbe per rizikingas ir nepaliktas pagrindinėje versijoje.

Mokomas kandidatų perrikiavimas. Vietoje aiškių taisyklių čia taikytas papildomai apmokytas modelis, kuris turėjo pats išmokti kandidatų tvarką. Tikrinta, ar toks sudėtingesnis metodas pranoks rankines taisykles. Praktikoje šios krypties rezultatai buvo jautresni nustatymams ir dažnai nusileido paprastesniam variantui.

Patikimumas ir atsakymo nepateikimas. Čia vertinta, kada sistema turėtų nepateikti vieno atsakymo, jei pirmi kandidatai per mažai skiriasi. Taip valdomas kompromisas tarp klaidų rizikos ir to, kokiai

užklausų daliai sistema atsako. Šie bandymai padėjo nustatyti ribas, kada pateikti atsakymą, o kada palikti atvejį peržiūrai.

Patikimumas (kalibravimas). Šioje grupėje tikrinta, ar pasitikėjimo balą galima paversti aiškiai suprantama tikimybe. Tai leidžia ribą, nuo kurios sistema neturėtų pateikti atsakymo, aiškinti paprasčiau ir nuosekliau taikyti skirtingoms grupėms. Tokiu būdu sprendimo elgsena tampa lengviau paaiškinama ir ne techniniam skaitytojui.

Kelių pavyzdžių režimas (laipsniškas scenarijus). Čia tikrinta, kaip greitai galima įtraukti naują retą klasę papildant paieškos rinkinį nedideliu naujų pavyzdžių kiekiu. Lygintos situacijos su 1, 3 ir 5 pavyzdžiais ir stebėta, kaip kinta rezultatas. Šios grupės išvados rodo, kiek sistema tinkama palapsniam plėtimui be pilno pertreniravimo ciklo.

Etikečių stambinimas. Šiuose bandymuose labai artimos klasės laikinai sujungtos į stambesnes grupes. Taip tikrinta, ar klaidos kyla dėl modelio silpnumo, ar dėl to, kad užduotis apibrėžta per smulkiai. Ši analizė padėjo atskirti modelio klaidas nuo paties klasių skirstymo problemos.

Pograpių analizė. Rezultatai buvo suskaidyti pagal duomenų pilnumą, pavyzdžiui, ar skersmuo žinomas. Tai leido aiškiai pamatyti, kur sistema veikia patikimai ir kur ją riboja duomenų trūkumas. Tokia analizė reikalinga tam, kad išvados būtų taikomos praktikoje, o ne tik bendrame vidurkyje.

1.15. Visų bandymų sąrašas

1.3 lentelėje pateikiamas visų bandymų sąrašas ir pagrindiniai jų nustatymai.

1.3 lentelė. Visų bandymų sąrašas

Kas pakeista	Svarbiausi nustatymai	Rezultatas (Tikslumas / makro F1)
Bazinis atskaitos taškas. Čia apmokytas pradinis modelis. Jis naudojamas kaip bendras palyginimo taškas visiems kitiems bandymams. Šioje grupėje svarbi ne didžiausia kokybė, o stabili bazė, su kuria lyginami vėlesni pakeitimai.		
Bazinio vaizdo modelio apmokymas (atskaitos taškas)	duomenys=Bazė (trūkstantumas pažymėtas)/bazė (ankstesnė)	vidurkis 0.8047 / 0.4501; intervalas 0.7875-0.8155 / 0.4184-0.4675
Bazinių metodų palyginimas. Čia lyginti patys paprasčiausi sprendimai. Vienu atveju sistema iškart grąžina vieną atsakymą, kitu atveju ieško panašiausių pavyzdžių pagal požymių vektorius. Taip matyti, kiek duoda vien vaizdas dar prieš papildomos informacijos perrikiavimą.		
Vienas atsakymas iš klasifikatoriaus	duomenys=Bazė (trūkstantumas pažymėtas)/bazė (ankstesnė); kandidatų sąrašo dydis=100/200; k=1; svoris(skait.)=0.5; svoris(data)=0.5; svoris(aut.)=0.0/0.5; beta=2.0/4.0	vidurkis 0.8957 / 0.6736; intervalas 0.7815-0.9452 / 0.4184-0.8096
Prototipų metodas (klasės vidurkis)	duomenys=bazė (ankstesnė); vaizdas=vektoriniai įterpiniai	0.8045 / 0.4795
Paieška pagal vaizdo panašumą	duomenys=bazė (ankstesnė); k=3; vaizdas=vektoriniai įterpiniai	vidurkis 0.8279 / 0.5021; intervalas 0.8258-0.8300 / 0.4973-0.5070
Klasifikatorius (softmax; vienas atsakymas)	kandidatų sąrašo dydis=500; vaizdas=tik vaizdas	0.9361 / 0.7376
Vaizdo koduotojo palyginimas (vienas atsakymas)	koduotojas=keli variantai	vidurkis 0.5806 / 0.3209; intervalas 0.2546-0.8016 / 0.0712-0.4912
Hierarchinis sprendimas. Čia klasė parenkama dviem žingsniais. Pirma nustatoma stambesnė grupė, po to joje ieškoma tikslios klasės. Taip tikrinta, ar toks skaidymas sumažina painiavą tarp labai artimų klasių.		

Hierarchinis sprendimas (2 etapų): pirma grupė, tada klasė	duomenys=bazė (ankstesnė)	vidurkis 0.8089 / 0.2691; intervalas 0.8088-0.8090 / 0.2688-0.2693
Pagrindinio varianto stabilumas. Čia paleistas tas pats galutinis sprendimo variantas, keičiant tik artimus nustatymus. Taip tikrinta, ar pagrindinė schema išlieka stabili ir ar rezultatas nesilaiko tik ant vieno atsitiktinai gero paleidimo.		
Pagrindinis variantas: vaizdo paieška + papildoma informacija + taisyklės	svoris(skait.)=0.5; svoris(data)=0.5; svoris(aut.)=0.0/0.5; beta=1.0/4.0; vaizdas=tik vaizdas	vidurkis 0.9391 / 0.7768; intervalas 0.9349-0.9427 / 0.7611-0.8026
Papildomos informacijos perrikiavimas. Čia paieška vyksta dviem žingsniais. Pirmą surenkami kandidatai pagal požymių vektorių panašumą, po to jų tvarka perskaičiuojama pagal svorį, skersmenį, metų laikotarpį ir kitus atitiktus požymius. Taip tikrinta, kiek duoda papildoma informacija, kai vien vaizdo neužtenka.		
Perrikiavimas pagal svorį, skersmenį ir metų laikotarpį	duomenys=Bazė (trūkstamumas pažymėtas)/bazė (ankstesnė); kandidatų sąrašo dydis=50/500; k=1/3; svoris(skait.)=0.2/0.5; svoris(data)=0.2/0.5; svoris(aut.)=0.0/0.2/0.5; beta=0.5/2.0/4.0	vidurkis 0.9286 / 0.7410; intervalas 0.8902-0.9492 / 0.5860-0.8190
Perrikiavimas pagal skaitinę informaciją (svoris / skersmuo)	duomenys=bazė (ankstesnė); kandidatų sąrašo dydis=50; k=1/3; svoris(skait.)=0.2; vaizdas=tik vaizdas/vektoriniai įterpiniai	vidurkis 0.8972 / 0.6174; intervalas 0.8913-0.9087 / 0.5848-0.6418
Perrikiavimas su metų intervalo ribojimu (patikra)	kandidatų sąrašo dydis=500	0.9430 / 0.7976
Metų intervalo ribojimo parametro (β) patikra	kandidatų sąrašo dydis=500; beta=2.0	0.9482 / 0.7945
Komponentų išjungimo patikra. Naudojama ta pati pagrindinė schema, tačiau po vieną išjungiamas konkretus komponentas. Pavyzdžiui, paliekamas tik vaizdo etapas arba atjungiamas viena papildomos informacijos dalis. Tokiu būdu kiekvieno komponento indėlis matuojamas atskirai, nekeičiant kitų dalių.		
Papildomos informacijos komponentų išjungimo patikra	kandidatų sąrašo dydis=500	vidurkis 0.9173 / 0.7205; intervalas 0.8860-0.9487 / 0.6303-0.8107
kNN balsavimas. Po vaizdo paieškos surenkami artimiausi kaimynai ir iš jų gaunamas galutinis balsas už klasę. Lygintos kelios taisyklės: paprastas daugumos balsas, svėrimas pagal atstumą ir svėrimas pagal panašumo balą. Taip pat tikrintas jautrumas kaimynų skaičiui ir panašumo skalės parametrui.		
kNN balsavimo taisyklės / svėrimo patikra	kandidatų sąrašo dydis=500; k=1; vaizdas=tik vaizdas	vidurkis 0.9255 / 0.7463; intervalas 0.8248-0.9482 / 0.5094-0.7993
Balsavimo / panašumo skalės (temperatūros) jautrumo patikra	kandidatų sąrašo dydis=500	0.9492 / 0.8189
kNN balsavimas: daugumos balsas	kandidatų sąrašo dydis=500; vaizdas=tik vaizdas	0.9464 / 0.7847
kNN balsavimas: svėrimas pagal atstumą	kandidatų sąrašo dydis=500; vaizdas=tik vaizdas	0.9482 / 0.7921
kNN balsavimas: svėrimas pagal panašumo balą	kandidatų sąrašo dydis=500; vaizdas=tik vaizdas	0.9492 / 0.8190
Kandidatų sąrašo dydis. Šioje grupėje keistas tik kandidatų kiekis, perduodamas į antrą etapą. Mažesnis sąrašas sumažina skaičiavimo kainą, bet gali neįtraukti teisingo kandidato, didesnis sąrašas didina triukšmą. Palyginimas atliktas su keliomis reikšmėmis, o kiti komponentai palikti tokie patys.		
Kandidatų sąrašo dydžio patikra	kandidatų sąrašo dydis=100/1000/200; vaizdas=tik vaizdas	vidurkis 0.9445 / 0.8095; intervalas 0.9373-0.9509 / 0.7981-0.8208

<p>Kandidatų sąrašo įvairovė. Po pradinės paieškos ribojama, kiek kartų ta pati klasė gali kartotis kandidatų sąrašė. Tikslas yra sumažinti dažnų klasių perteklių ir palikti daugiau skirtingų alternatyvų prieš perrikiavimą. Taip tikrinama, ar įvairesnis kandidatų sąrašas padeda teisingą kandidatą perkelti į pirmą vietą.</p>		
Kandidatų sąrašo įvairovės didinimas (ribojant vienos klasės pasikartojimą)	kandidatų sąrašo dydis=500; vaizdas=tik vaizdas	vidurkis 0.9493 / 0.8206; intervalas 0.9492-0.9494 / 0.8190-0.8221
<p>Paieškos patobulinimai. Pritaikyti greiti pakeitimai po pirmos paieškos, nepertraukiant mokymo. Čia tikrintos trys idėjos. Pirmą, užklauso išplėtimas. Po pirmos paieškos paaimami keli viršutiniai kandidatai, apskaičiuojamas jų požymių vektorių vidurkis, ir ši vidutinė reikšmė pridedama prie pradinio užklauso vektoriaus. Tada paieška paleidžiama dar kartą su atnaujinta užklausa, tikintis, kad viršuje atsiras daugiau teisingos klasės kandidatų. Antra, kaimynystės persidengimas. Kandidatai kilsteliami, jei jų kaimynai panašūs į užklauso kaimynus. Trečia, kandidatų vietų sujungimas. Kiekvienas kandidatas gauna vietą keliuose atskiruose sąrašuose, o galutinė vieta skaičiuojama pagal šias vietas. Tai reiškia, kad lyginamos ne pačios balų reikšmės, o kandidatų eilės numeriai skirtinguose sąrašuose.</p>		
Užklauso išplėtimas (užklauso vektorius papildomas artimiausių kandidatų vidurkiu)	kandidatų sąrašo dydis=500; k=20; vaizdas=tik vaizdas	vidurkis 0.9457 / 0.7900; intervalas 0.9443-0.9484 / 0.7894-0.7903
Kandidatų vietų sujungimas (RRF) vietoje balų sumavimo	kandidatų sąrašo dydis=500; k=0; vaizdas=tik vaizdas	0.9256 / 0.6479
Kaimynų persidengimo (konteksto) panaudojimas reitingui	vaizdas=tik vaizdas	0.9403 / 0.7797
<p>Paieškos rinkinio mažinimas. Prieš paiešką sumažintas palyginimo rinkinys, kad sumažėtų vykdymo kaina. Tikslas buvo įvertinti, ar greitėjimas ne per brangiai kainuoja kokybės prasme. Bandyti keli mažinimo principai, pavyzdžiui, palikti arčiausius klasės centrai arba riboti tik dažniausias klases.</p>		
Galerijos mažinimas: paliekami artimiausi klasės centrai pavyzdžiai	kandidatų sąrašo dydis=500	0.9067 / 0.6868
Galerijos mažinimas: atsitiktinai paliekama dalis pavyzdžių kiekvienoje klasėje	kandidatų sąrašo dydis=500	vidurkis 0.9346 / 0.7137; intervalas 0.9186-0.9460 / 0.6431-0.7821
Galerijos mažinimas: ribojamos tik dažnos klasės (head-cap)	kandidatų sąrašo dydis=500	vidurkis 0.9321 / 0.7975; intervalas 0.9225-0.9417 / 0.7895-0.8056
<p>Papildomas laukas: kaldinimo vieta. Prie įprasto perrikiavimo įtraukta kaldinimo vieta. Vienuose bandymuose ji naudota kaip minkštas balo priedas, kituose kaip griežtas atmetimo kriterijus. Taip patikrinta, ar šis laukas stabiliai padeda, kai dalyje įrašų jis nepilnas arba nevienodas.</p>		
Kaldinimo vietos lauko įtraukimas	duomenys=bazė + kaldinimo vieta; kandidatų sąrašo dydis=200/500; k=3; svoris(skait.)=0.5; svoris(data)=0.5; svoris(aut.)=0.0; svoris(kaldinimo vieta)=0.0/0.05/0.2	vidurkis 0.9415 / 0.7612; intervalas 0.9245-0.9505 / 0.6937-0.7932
Kaldinimo vietos laukas kaip filtras (griežtas atmetimas)	duomenys=bazė + kaldinimo vieta; kandidatų sąrašo dydis=500; vaizdas=tik vaizdas	vidurkis 0.9114 / 0.7783; intervalas 0.8982-0.9247 / 0.7754-0.7813
<p>Griežtas atmetimas. Po pradinės paieškos dalis kandidatų atmetama iš karto, jei nesutampa papildoma informacija. Toks kelias atrodo paprastas, bet jis pavojingas, kai laukai nepilni ar netikslūs. Čia tikrinta, ar ankstyvas atmetimas nepašalina teisingo kandidato.</p>		
Griežtas kandidatų atmetimas pagal papildomą informaciją (filtras)	vaizdas=tik vaizdas	vidurkis 0.9476 / 0.8147; intervalas 0.9452-0.9492 / 0.8096-0.8190
<p>Mokomas perrikiavimas. Po pradinės paieškos kiekviena užklausa ir kandidatas paverčiami požymių rinkiniu, kurį vertina atskirai apmokytas modelis. Tada kandidatai surikiuojami pagal jo išvestą balą. Taip tikrinta, ar sudėtingesnis antras etapas duoda daugiau naudos negu paprastos taisyklės.</p>		

Mokomas perrikiavimas: logistinė regresija (2 etapas)	duomenys=bazė + kaldinimo vieta; kandidatų sąrašo dydis=100/200; beta=4.0; vaizdas=tik vaizdas	vidurkis 0.9280 / 0.7041; intervalas 0.9150-0.9355 / 0.6095-0.7496
Mokomas perrikiavimas: medžių ansamblis (2 etapas)	duomenys=bazė + kaldinimo vieta; kandidatų sąrašo dydis=100/200/500; beta=4.0; vaizdas=tik vaizdas	vidurkis 0.9311 / 0.7003; intervalas 0.9274-0.9344 / 0.6772-0.7193
Mokomas perrikiavimas: porinis verifikatorius	kandidatų sąrašo dydis=200/500; vaizdas=tik vaizdas	vidurkis 0.8885 / 0.5990; intervalas 0.8621-0.9335 / 0.4949-0.7650
Mokomas perrikiavimas: LightGBM LambdaRank	kandidatų sąrašo dydis=50/500; vaizdas=tik vaizdas	vidurkis 0.9347 / 0.7287; intervalas 0.9306-0.9389 / 0.7242-0.7332
Vektorinių įterpinių mokymo bandymai. Pakeistas pats vaizdo modelio mokymo tikslas, tai yra kaip formuojama požymių erdvė. Po kiekvieno tokio mokymo paleistas tas pats paieškos ir perrikiavimo vertinimas kaip bazėje, kad skirtumas būtų siejamas tik su mokymo tikslu. Tikrinti keli variantai. Pagrindinis porinis mokymas traukia tos pačios klasės poras arčiau ir stumia skirtingų klasių poras toliau. FastAP variante mokymo metu tiesiogiai skatinama geresnė paieškos tvarka viršutinėse pozicijose. ProxyNCA variante kiekviena klasė turi savo atstovą, o modelis mokomas užklausa artinti prie teisingos klasės atstovo ir tolinti nuo kitų klasių atstovų. Taip aiškiai patikrinama, ar nauda kyla būtent iš mokymo tikslo pakeitimo.		
Vektorinių įterpinių mokymo variantas (mokymas pagal poras / netolygus)	kandidatų sąrašo dydis=200/500; svoris(skait.)=0.5; svoris(data)=0.5; vaizdas=tik vaizdas/vektoriniai įterpiniai	vidurkis 0.9406 / 0.7632; intervalas 0.9379-0.9432 / 0.7608-0.7656
Vektorinių įterpinių mokymas su FastAP (rikiavimo idėja)	kandidatų sąrašo dydis=500; vaizdas=vektoriniai įterpiniai	0.9459 / 0.7335
Vektorinių įterpinių mokymas su ProxyNCA (klasių atstovai)	kandidatų sąrašo dydis=500; vaizdas=vektoriniai įterpiniai	0.9427 / 0.7572
Dviejų pusių informacija. Lyginta, kaip veikia tik priekinė pusė, tik galinė pusė ir abiejų pusių derinys. Papildomai tikrintas fiksuotas sujungimo svoris, kuris leidžia labiau pasitikėti viena puse nei kita. Taip tikrinama, ar skirtingi pusių naudojimo režimai nepablogina atskyrimo tarp artimų klasių.		
Naudojama tik priekinė pusė (dalis išjungimo patikra)	kandidatų sąrašo dydis=500; vaizdas=tik priekinė pusė	0.9334 / 0.7426
Naudojama tik galinė pusė (dalis išjungimo patikra)	kandidatų sąrašo dydis=500; vaizdas=tik galinė pusė	0.9329 / 0.7520
Dviejų pusių sujungimo svorio patikra (α)	kandidatų sąrašo dydis=500; vaizdas=tik vaizdas	vidurkis 0.9459 / 0.7956; intervalas 0.9436-0.9478 / 0.7806-0.8087
Patikimumas. Įvertintas režimas, kai sistema daliai užklausių nepateikia vieno atsakymo. Toks sprendimas įjungiamas pagal pasitikėjimo signalą, dažniausiai pagal skirtumą tarp pirmo ir antro kandidato balų. Taip valdoma pusiausvyra tarp mažesnės klaidų rizikos ir to, kiek užklausių sistema atsako. Čia lyginta, kaip keičiasi rezultatas, kai slenkstis griežtinamas arba silpninamas.		
Patikimumo veikimo santrauka (kompromisas tarp to, kiek užklausių sistema atsako, ir kokybės)	—	0.9452 / 0.8096
Riba, nuo kurios sistema neturėtų pateikti atsakymo	duomenys=Bazė + fono šalinimas/Bazė + triukšmo mažinimas + CLAHE/Bazė + vaizdo kontrasto didinimas (CLAHE); kandidatų sąrašo dydis=500; k=1; vaizdas=tik vaizdas	vidurkis 0.9275 / 0.7427; intervalas 0.7718-0.9492 / 0.2617-0.8190
Patikimumas ir kalibravimas. Čia vidinis patikimumo balas paverčiamas į tikimybę nuo 0 iki 1. Tada tikrinama, ar pagal tokią tikimybę galima aiškiau valdyti, kada sistema neturėtų pateikti atsakymo. Šis bandymas svarbus todėl, kad tikimybinę ribą lengviau paaiškinti negu neapdorotą balų skirtumą.		

Patikimumo balo pavertimas į tikimybę (S formos kreivė)	kandidatų sąrašo dydis=500; vaizdas=tik vaizdas	vidurkis 0.9492 / 0.8190; intervalas 0.9492-0.9492 / 0.8190-0.8190
Atsparumo patikra. Čia validacijoje dirbtinai paslėpta dalis papildomos informacijos laukų. Taip imituojamos realios situacijos, kai papildoma informacija nepilna. Po to paleista ta pati schema ir įvertinta, kiek stipriai nuo to krenta rezultatai.		
Dirbtinai paslėpta papildoma informacija (robustumo patikra)	vaizdas=tik vaizdas	vidurkis 0.9492 / 0.8190; intervalas 0.9492-0.9492 / 0.8190-0.8190
Nežinomų reikšmių suvienodinimas (patikra)	kandidatų sąrašo dydis=500; vaizdas=tik vaizdas	0.9492 / 0.8190
Laipsniškas naujų klasių įtraukimas. Nauja reta klasė įtraukta be viso modelio pertreniruotės. Į paieškos rinkinį pridedami keli naujos klasės pavyzdžiai ir tikrinama, kaip keičiasi jos atpažinimas. Taip matuojamas praktinis kelių pavyzdžių scenarijus, kai sistemos plėtra turi būti greita.		
Naujos (retos) klasės pridėjimas su keliais pavyzdžiais (be pertreniruotės)	duomenys=bazė + kaldinimo vieta; kandidatų sąrašo dydis=100/200/50/500; svoris(kaldinimo vieta)=0.0/0.2; vaizdas=tik vaizdas	vidurkis 0.9451 / 0.7916; intervalas 0.9393-0.9505 / 0.7715-0.8190
Etikečių stambinimas. Labai artimos klasės laikinai sujungtos į stambesnes grupes ir vertinimas atliktas su šiomis grupėmis. Tai leidžia atskirti dvi klaidų priežastis: modelio ribas ir per smulkų klasių skaidymą duomenyse. Jei po sujungimo rezultatas ženkliai gerėja, dalis klaidų kyla būtent dėl pernelyg smulkių ribų.		
Etikečių sujungimas: dvi taisyklės (nominalas + valdovas ir drachmų šeima)	kandidatų sąrašo dydis=500	vidurkis 0.9612 / 0.8246; intervalas 0.9452-0.9772 / 0.8096-0.8397
Etikečių sujungimas: nominalas + valdovas	—	0.9452 / 0.8096
Etikečių sujungimas: drachmų šeima	—	0.9452 / 0.8096
Etikečių sujungimas: pagal supainiojimus (patikra)	—	0.9492 / 0.8190
Pograpių analizė. Tas pats modelis vertintas atskirai pagal duomenų pilnumą, pavyzdžiui, kai skersmuo žinomas ir kai jis nežinomas. Taip matyti, ar klaidas labiau lemia pats modelis, ar papildomos informacijos trūkumas.		
Pograpių analizė (pvz., skersmuo žinomas / nežinomas)	kandidatų sąrašo dydis=500	0.9492 / 0.8190
Vaizdo koduotojo palyginimas. Čia keistas tik vaizdo koduotojas, o visa kita palikta taip pat. Visi variantai vertinti su tuo pačiu kandidatų sąrašu ir tuo pačiu antru etapu. Taip skirtumas labiau priklauso nuo paties vaizdo modelio, o ne nuo kitų dalių.		
Vaizdo koduotojo palyginimas (paieška + perrikiavimas)	kandidatų sąrašo dydis=500; koduotojas=keli variantai	vidurkis 0.9369 / 0.7428; intervalas 0.9162-0.9527 / 0.6727-0.8258
Neaiškių atvejų sprendimas. Kai pirmi kandidatai turi labai panašius balus, sistema gali laikinai išplėsti tikrinamų kandidatų kiekį. Čia tikrinta, nuo kokios ribos toks papildomas tikrinimas pradeda duoti naudą ir kada jis tik įneša daugiau triukšmo.		
Adaptyvus k (kai keli variantai labai panašūs)	kandidatų sąrašo dydis=500	vidurkis 0.9462 / 0.8115; intervalas 0.9462-0.9462 / 0.8115-0.8115
Adaptyvus k: ribų / slenksčių patikra	k=3/5; vaizdas=tik vaizdas	vidurkis 0.9477 / 0.8047; intervalas 0.9462-0.9492 / 0.7904-0.8190
Mokymo nustatymų patikra. Tikrinama, ar duomenų augmentacija mokymo metu pagerina apibendrinimą. Mokymo režime taikomi vaizdo pakeitimai, o vertinime naudojama ta pati vertinimo tvarka kaip bazėje. Tai leidžia patikrinti, ar papildomas mokymo sudėtingumas duoda realų praktinį efektą.		
Vaizdų keitimas mokymo metu (augmentacija) – patikra	kandidatų sąrašo dydis=500	0.9245 / 0.6937

<p>Statistikos pagrindu perskaičiuoti balai. Pirma iš mokymo duomenų suskaičiuota, kokios papildomos informacijos reikšmės dažniausios kiekvienoje klasėje. Tada kiekvienam kandidatui apskaičiuotas papildomas balas pagal jo klasę ir jo papildomos informacijos reikšmes. Jei kandidato reikšmės labiau atitinka tai klasei būdingą pasiskirstymą, jo balas padidinamas; jei neatitinka, balas padidinamas mažiau arba mažinamas. Po to kandidatai vėl surikiuojami ir vertinama, ar toks perskaičiavimas pagerina galutinį atsakymą.</p>		
<p>Statistinis balo perskaičiavimas pagal klasę ir papildomą informaciją</p>	<p>duomenys=bazė + kaldinimo vieta; vaizdas=tik vaizdas</p>	<p>0.9482 / 0.7932</p>
<p>Klasės tikėtinumo balo perskaičiavimas pagal papildomą informaciją (patikra)</p>	<p>kandidatų sąrašo dydis=500; vaizdas=tik vaizdas</p>	<p>0.9423 / 0.7816</p>

Ši lentelė surenka visus bandymus į vieną vietą. Joje matyti, kas buvo keičiama, kokie nustatymai naudoti ir koks rezultatas gautas. Ji skirta greitai orientacijai prieš skaitant išsamias kiekvienos bandymų grupės išvadas.

1.16. Vien vaizdo informacijos pakankamumas

1.4 lentelėje pateikiami bandymai, kurie rodo, kiek šiam uždaviniui pakanka vien vaizdo informacijos.

1.4 lentelė. Vien vaizdo informacijos pakankamumo rezultatų suvestinė

Variantas	Pagrindiniai rezultatai	Išvada
Softmax klasifikatorius	makro F1 0,4258 tikslumas 0,7784	Vieno atsakymo bazė silpna retoms klasėms.
Prototipai	makro F1 0,4556 retų klasių makro F1 0,3790	Vien vaizdo neužtenka.
kNN be papildomos informacijos	makro F1 0,4630 retų klasių makro F1 0,3687	Vien vaizdo neužtenka.
kNN su papildomos informacijos perikiavimu	makro F1 0,8190 atkūrimas@5 0,9787	Papildoma informacija duoda aiškų pagerėjimą.

Rezultatai rodo, kad vien vaizdo informacijos šiam uždaviniui per mažai, ypač retoms klasėms. Pagrindinis pagerėjimas atsiranda tik tada, kai po paieškos įtraukiama papildoma informacija.

1.17. Kandidatų sąrašo formavimo strategijos

1.5 lentelėje pateikiami kandidatų sąrašo formavimo strategijų rezultatai.

1.5 lentelė. Kandidatų sąrašo formavimo strategijų rezultatų suvestinė

Variantas	Pagrindiniai rezultatai	Išvada
AQE tipo užklauso išplėtimas	makro F1 0,7903 atkūrimas@5 0,9750	Stabilaus pagerėjimo nedavė.
Kontekstinis perikiavimas	makro F1 0,7894	Stabilaus pagerėjimo nedavė.

	atkūrimas@5 0,9761	
Kandidatų įvairovės didinimas	makro F1 0,8221 atkūrimas@1 0,9671	Davė nedidelį, bet nuoseklų pagerėjimą.
Mažesnis pradinis sąrašas 1 000	makro F1 0,8190	Įvairovės nauda išnyko.

Iš paprastų paieškos taisyklių realią naudą davė tik kandidatų įvairovės didinimas. Kiti greiti pataisymai šiame uždavinyje rezultatą silpnino arba nekeitė.

1.18. Vaizdo požymių gerinimo bandymai

1.6 lentelėje pateikiami vaizdo požymių gerinimo bandymų rezultatai.

1.6 lentelė. Vaizdo požymių gerinimo bandymų rezultatų suvestinė

Variantas	Pagrindiniai rezultatai	Išvada
FastAP	makro F1 0,7335	Žemiau bazinio sprendimo 0,8190.
ProxyNCA	makro F1 0,7572	Žemiau bazinio sprendimo 0,8190.
PaCo tipo variantas	makro F1 0,7656	Žemiau bazinio sprendimo 0,8190.
SupCon+CE	makro F1 0,7632 be perrikiavimo 0,5143	Galutinio sprendimo nepagerino.

Ši lentelė parodo, kad kitoks mokymo būdas savaime negarantuoja geresnio galutinio rezultato. Nė vienas iš šių variantų neaplenkė bazinio sprendimo su paprastu perrikiavimu.

1.19. Mokomas modelis kandidatų tvarkai keisti ir jo stabilumas

1.7 lentelėje pateikiami mokomo modelio kandidatų tvarkai keisti rezultatai.

1.7 lentelė. Mokomo modelio kandidatų tvarkai keisti rezultatų suvestinė

Variantas	Pagrindiniai rezultatai	Išvada
LightGBM LambdaRank, 50 kandidatų	makro F1 0,7242	Bazinio sprendimo neaplenkė.
LightGBM LambdaRank, 500 kandidatų	makro F1 0,7332	Bazinio sprendimo neaplenkė.
Porinis verifikatorius	makro F1 0,5412 atkūrimas@1 0,9054	Vieno atsakymo kokybė per silpna.
Porinis verifikatorius su sunkiomis neigiamomis	makro F1 0,6023	Rezultatas pagerėjo, bet liko per žemas.
Verifikatorius kaip papildomas signalas	makro F1 0,7650	Taisyklių neaplenkė net kaip papildomas signalas.

Rezultatai rodo, kad mokomas modelis kandidatų tvarkai keisti šiame duomenų variante buvo jautresnis nustatymams ir neaplenkė paprasto taisyklinio perrikiavimo. Dėl to pagrindiniame variante paliktas aiškiau paaiškinamas bazinis sprendimas.

1.20. Patikimumo ir neatsakymo analizė

1.8 lentelėje pateikiami patikimumo ir neatsakymo analizės rezultatai.

1.8 lentelė. Patikimumo ir neatsakymo analizės rezultatų suvestinė

Variantas	Pagrindiniai rezultatai	Išvada
Balų skirtumo signalas	AURC 0,0278	Veikia kaip bazinis patikimumo signalas, bet silpnesnis.
Kalibruota tikimybė	AURC 0,0143 sistema atsako į 88,20 % užklausų atkūrimas@5 0,9813	Geriau rikiuoja riziką ir leidžia aiškiau valdyti, kada sistema atsako į 88,20 % užklausų.

Kalibruota tikimybė pasirodė geresnė už vien balų skirtumą. Vis dėlto, taikant 0,95 slenkstį, atsakymų dalis labiau sumažėja vidutinėms ir retoms klasėms, todėl šį sprendimą reikia vertinti kartu su tuo, kiek atvejų sistema pasilieka sau atsakyti.

1.21. Mastelio ir stabilumo kompromisai

1.9 lentelėje pateikiami mastelio ir stabilumo kompromisų rezultatai.

1.9 lentelė. Mastelio ir stabilumo kompromisų rezultatų suvestinė

Variantas	Pagrindiniai rezultatai	Išvada
500 kandidatų sąrašas	makro F1 0,8190 atkūrimas@5 0,9787	Bazinis kompromisas tarp kokybės ir kainos.
200 kandidatų sąrašas	makro F1 0,8096 atkūrimas@5 0,9793	Skaičiavimo mažiau, bet vieno atsakymo kokybė silpnesnė.
Retinimas iki 50 % pagal centroidą	makro F1 0,6868 atkūrimas@5 0,9437	Kokybės kritimas per didelis.
Dažnų klasių riba max 200	makro F1 0,8056 mokymo rinkinys sumažintas iki 2002	Atsargesnis kompromisas.
Dažnų klasių riba max 100	makro F1 0,7895 mokymo rinkinys sumažintas iki 1316	Daugiau taupo, bet daugiau kainuoja kokybės.

Rezultatai rodo, kad skaičiavimo kainą galima mažinti, bet ne bet kokia kaina. Didesnis kandidatų sąrašas ir atsargesnis dažnų klasių ribojimas duoda priimtinesnį kompromisą negu agresyvus retinimas.