VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ STUDIJŲ PROGRAMA

Vaizdų panašumo vertinimas neapibrėžtose dalykinėse srityse

Evaluation of image similarity in undefined domains

Magistro baigiamasis darbas

Atliko:	Augustinas Makevičius	(parašas)
Darbo vadovas:	j. asist. Boleslovas Dapkūnas	(parašas)
Recenzentas:	asist. dr. Tomas Plankis	(parašas)

Vilnius – 2023

Padėka

Darbo autorius dėkoja Vilniaus universiteto Matematikos ir informatikos fakulteto Informacinių technologijų atviros prieigos centrui už suteiktus HPC išteklius šio darbo skaičiavimams atlikti.

Santrauka

Duomenų kiekiams nuolatos augant, ieškoti panašių vaizdų rankiniu būdu nebėra optimalu. Reikalingas panašumo vertinimo automatizavimas, leidžiantis įvertinti vaizdų panašumą per trumpą laiko tarpą ir sumažinantis vertinimo subjektyvumą. Šio darbo tikslas – pasiūlyti naują vaizdų, kurių dalykinės sritys yra neapibrėžtos, panašumo vertinimo metodą. Tiriami jau egzistuojantys panašumo vertinimo metodai (GIST, MultiGrain, HOW, „VisionForce“ ir klasifikacijos užduočiai pritaikyti metodai) ir naujai pasiūlytas vaizdų transformatoriais ir ketvertų tikslo funkcija paremtas metodas. Metodų vertinimas atliekamas naudojant „DISC21“ ir „Google Landmarks Dataset v2“ (GLDv2) duomenų rinkinius. Geriausią preciziškumą, naudojant „DISC21“ duomenų rinkinį, parodė „VisionForce“, HOW ir autoriaus pasiūlytas metodas, o klasifikacijos modeliais paremtas metodas parodė prasčiausius rezultatus. Geriausią preciziškumą, naudojant „LAND“ duomenų rinkinį, parodė pasiekia HOW ir autoriaus pasiūlytas metodas. Ištestavus modelių našumą pastebėta, kad greičiausiai veikia GIST metodas. Geriausius rezultatus parodę „HOW“ ir „VisionForce“ metodai pasižymi mažiausiu savybių išgavimo greičiu. Autoriaus pasiūlytas metodas yra antroje vietoje našumo atžvilgiu.

Raktiniai žodžiai: vaizdų panašumas, vaizdų transformatoriai, ketvertų tikslo funkcija.

Santrauka anglų kalba (*Summary*)

As data volumes continue to grow, it is no longer optimal to manually search for similar images in large image datasets. Automation of the evaluation of image similarity is needed to evaluate the similarity in a short timeframe and to reduce the subjectivity of the evaluation. The aim of this work is to propose a new method for the evaluation of image similarity in undefined subject domains. Existing similarity evaluation methods (GIST, MultiGrain, HOW, VisionForce and methods adapted to the classification task) and a newly proposed method based on vision transformers and the quadruplet loss function are investigated. The methods are evaluated using the „DISC21“ and „Google Landmarks Dataset v2“ (GLDv2) datasets. VisionForce, HOW and the author's proposed method showed the best precision on the „DISC21“ dataset, while the method based on classification models showed the worst results. The best precision on the „GLDv2“ dataset was achieved by the HOW method and the author's proposed method. The GIST method was found to be the fastest when the performance of the models was tested. The best performing HOW and VisionForce methods have the lowest feature extraction speed. The author's proposed method is ranked second in terms of performance.

Keywords: image similarity, vision transformers, quadruplet loss function.

TURINYS

Įvadas

Panašūs objektai – tai objektai, turintys tokių pat ypatybių ar bruožų. Tačiau panašumas yra abstrakti sąvoka. Kiekvienas asmuo panašumą gali suprasti kitaip. Tam įtakos daro lyginimo požymiai, objektų kokybė, lygintojo asmenybė.

(a) Žalios spalvos obuoliai (b) Raudonos spalvos obuoliai

1 pav. Obuolių vaizdai [KRA+20]

Pavyzdys – 1 paveikslėlyje lyginami vaizdai yra panašūs (vaizduojami obuoliai), tačiau vaizdai skiriasi (skirtinga objektų spalva, kiekis, laikymo vieta). Žmogus lygina pasitelkdamas jam suprantamus ir pastebimus subjektyvius požymius.

Duomenų kiekiams nuolatos augant [GR12], ieškoti panašių vaizdų rankiniu būdu nebėra optimalu. Reikalingas panašumo vertinimo automatizavimas, leidžiantis įvertinti vaizdų panašumą per trumpą laiko tarpą. Be to, panašumo vertinimo automatizavimas leidžia sumažinti subjektyvaus vertinimo (žmogus vertina jam suprantamus požymius) įtaką, taip padidinant vertinimo tikslumą ir patikimumą (pridedama objektyvumo).

Jei turėtume tikslius lyginimo požymius (pvz. obuolio spalva), galėtume apmokyti vaizdų klasifikacijos modelį ir atliekant prognozes skirstyti vaizdus į atitinkamas kategorijas. Tačiau problemos iškyla, jei panašumo kriterijai nėra tiksliai aprašyti, jų yra itin daug ar neturime tiksliai sužymėtų duomenų klasifikacijos modelio apmokymui. Tuomet pravartu naudoti platesnį vaizdų panašumo vertinimą, skirtą neapibrėžtų dalykinių sričių vaizdams.

Vaizdų panašumo vertinimas naudojamas daugelyje sričių:

- Atvirkštinė vaizdų paieška – tai informacijos paieškos būdas, kai paieškos varikliui paduodamas vaizdas, o vartotojui grąžinama naudinga su vaizdu susijusi informacija. Tai gali būti informacija apie vaizdo autorių, interneto svetaines, kuriose vaizdas patalpintas, aukštesnės raiškos vaizdo variantus ir pan. [HLA+19]

- Įžymių vietų atpažinimas – naudojantis vaizdų panašumo vertinimu, galime tiksliai nusakyti kokią įžymią vietą matome vaizde. [HS20]

- Vaizdų grupavimas – naudodami vaizdų panašumo vertinimą, galime sugrupuoti didelį duomenų rinkinį į mažesnes grupes – klasterius, be būtinybės ruošti duomenis, juos klasifikuoti bei apmokyti klasifikavimo modelį. [RPA14]

Atsižvelgiant į vaizdų panašumų vertinimo poreikį įvairiose industrijos srityse, nuspręsta ištirti vaizdų panašumo vertinimo būdus, palyginti jų reikalavimus bei rezultatus ir pasiūlyti naują vertinimo metodą.

Darbo metu bus apžvelgiami klasikiniai kompiuterinės regos panašumų vertinimo metodai bei dirbtiniais neuroniniais tinklais (giliaisiais neuroniniais tinklais bei vaizdų transformatoriais) pagrįsti metodai.

Darbo tikslas – pasiūlyti naują vaizdų, kurių dalykinės sritys yra neapibrėžtos, panašumo vertinimo metodą.

Darbo uždaviniai:

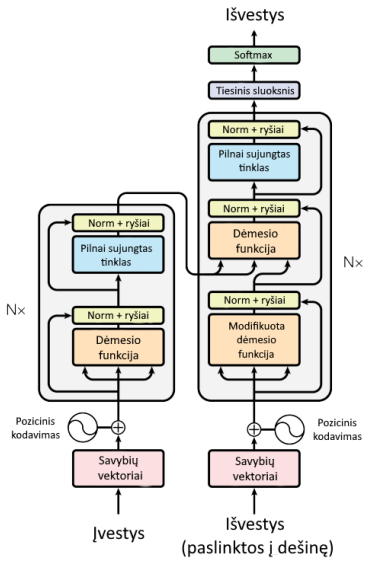
1. Apibrėžti vaizdų panašumo vertinimo metodų vertinimo bei lyginimo kriterijus.
2. Įgyvendinti vaizdų panašumo vertinimą keliais skirtingais metodais.
3. Pasiūlyti naują vaizdų panašumo vertinimo metodą.
4. Palyginti jau egzistuojančių bei pasiūlyto vaizdų panašumo vertinimo metodų rezultatus.

1. Mašininio mokymo metodai

Darbo metu apžvelgti vaizdų panašumo vertinimo metodai yra paremti įvairiomis modelių architektūromis, matematinėmis operacijomis, vektorių panašumo vertinimo metodais. Šiame skyriuje apžvelgiame jų ypatumus.

1.1. Transformatoriai

Rekurentinių neuroninių tinklų modeliai atlieka skaičiavimus pagal įvesties ir išvesties simbolių pozicijas. Suderinus pozicijas su skaičiavimo laiko etapais, sukuriama paslėptų būsenų seka H, kurios pozicija $h_i \in H$ priklauso nuo ankstesnės paslėptos būsenos h_{i-1} ir pozicijos i įvesties. Ši tinklo savybė neleidžia išlygiagretinti mokymo duomenų apdorojimo, o tai tampa itin svarbu apdorojant ilgesnes sekas. Transformatorius – tai modelio architektūra, kurioje vengiama pasikartojimo, o vietoj to, norint nustatyti globalias priklausomybes tarp įvesties ir išvesties, pasikliaunama vien tik dėmesio mechanizmu [VSP+17]. Modelio architektūrą galima matyti 2 paveikslėlyje.



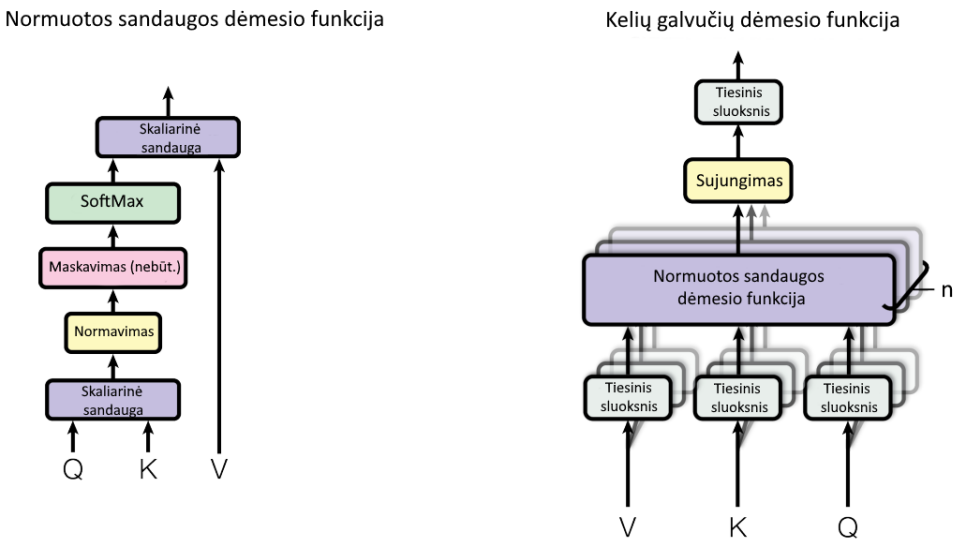
2 pav. Transformatorių architektūra [VSP+17]

Kaip ir daugelis geriausius rezultatus rodančių neuroninių sekos perdavimo modelių, transformatoriai pasižymi struktūra sudaryta iš koduotojo ir dekoduotojo. Koduotojas paverčia įvesties simbolių seką $x = (x_1,...,x_n)$ į seką savybių vektorių $z = (z_1,...,z_n)$. Dekoduotojas paverčia seką z į išvesties seką $y = (y_1,...,y_n)$, apdorodamas kiekvieną elementą iš eilės. Kiekviename žingsnyje modeliai yra autoregresiniai – t. y. naudojantys praeitų žingsnių išvestis, kaip papildomą įvestį naujos išvesties generavimui.

1.1.1. Dėmesio funkcija

Dėmesio funkciją galima apibūdinti kaip užklausos (Q) ir raktų (K) bei verčių (V) porų rinkinio pavertimą į išvestį, kur užklausa, raktai, vertės ir išvestis yra vektoriai. Išvestis apskaičiuojama kaip svertinė verčių suma, kai kiekvienai vertei priskiriamas svoris yra apskaičiuojamas pagal

suderinamumo funkciją priklausančią nuo užklausos ir vertės rakto. Transformatoriuose naudojama kelių galvučių dėmesio funkcija, kurią sudaro keli normuotos sandaugos dėmesio funkcijų sluoksniai (funkcijų sudedamąsias dalis matome 3 paveikslėlyje).



3 pav. Dėmesio funkcijos [VSP+17]

Normuotos sandaugos dėmesio funkcijos įvestį sudaro d_K dydžio užklausos ir raktų vektoriai bei d_V dydžio verčių vektoriai. Apskaičiuojame užklausos ir visų raktų skaliarines sandaugas, kiekvieną jų padalijame iš $\sqrt{d_K}$ ir, norėdami gauti verčių svorius, taikome „SoftMax“ funkciją. Proceso formulė pateikiama 1 formulėje.

$$D(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V \tag{1}$$

Vietoj to, kad su d_{model} dimensijų užklausomis, raktais ir vertėmis būtų apskaičiuojama viena dėmesio funkcija, Vaswani et al. [VSP+17] pastebi, kad naudinga n kartų tiesiškai suprojektuoti užklausas, raktus ir vertes (naudojantis skirtingomis projektavimo funkcijomis) į d_K, d_K ir d_V dimensijas. Kiekvienai iš šių projektuojamų užklausų, raktų ir verčių versijų lygiagrečiai apskaičiuojame dėmesio funkciją, gaudami n išvesties vektorių. Šie vektoriai sujungiami ir dar kartą suprojektuojami, taip gaunant galutinę dėmesio funkcijos išvestį. Kelių galvučių dėmesio funkcijos formulė pateikiama 2 formulėje.

$$KGD(Q,K,V) = concat\left(D\left(QW_1^Q, KW_1^K, VW_1^V\right),...,D\left(QW_n^Q, KW_n^K, VW_n^V\right)\right)W^O \tag{2}$$

Čia $W_i^Q \in \mathbb{R}^{d_{model}\times d_K}, W_i^K \in \mathbb{R}^{d_{model}\times d_K}, W_i^V \in \mathbb{R}^{d_{model}\times d_V}, W^O \in {}^{n\times d_V\times d_{model}}$ – parametrų matricos.

Transformatoriai naudoja kelių galvučių dėmesio funkciją trim skirtingais būdais:

- „Koduotojo-dekoduotojo dėmesio“ sluoksniuose užklausos gaunamos iš ankstesnio dekoduotojo sluoksnio, o atminties raktai ir vertės – iš koduotojo sluoksnio išvesties. Tai leidžia kiekvienai dekoduotojo pozicijai atkreipti dėmesį į visas įvesties sekos pozicijas, imituojant tipinius koduotojo-dekoduotojo dėmesio mechanizmus.

- Koduotojo savaiminio dėmesio sluoksniuose visi raktai, vertės ir užklausos gaunamos iš tos pačios vietos – ankstesnio koduotojo sluoksnio išvesties. Kiekviena koduotojo pozicija gali būti susijusi su visomis ankstesnio koduotojo sluoksnio pozicijomis.

- Dekoduotojo savaiminio dėmesio sluoksniai leidžia kiekvienai dekoduotojo pozicijai kreipti dėmesį į visas dekoduotojo pozicijas iki tos pozicijos imtinai. Tai daroma norint išsaugoti autoregresijos savybę ir yra įgyvendinama normuotos sandaugos dėmesio funkcijos skaičiavimo metu („SoftMax“ funkcijos įvesties vertės, kurios aprašo ryšius su tolimesnėmis pozicijomis, yra prilyginamos $-\infty$).

1.1.2. Pilnai sujungti tiesiniai tinklai

Be dėmesio funkcijų, transformatoriuose naudojami pilnai sujungti tiesiniai tinklai, kurie yra vienodai taikomi kiekvienai transformatoriaus pozicijai atskirai. Juos sudaro dvi tiesinės transformacijos su „ReLU“ aktyvacija tarp jų. Šio tinklo funkciją matome 3 formulėje.

$$FFN(x) = max\left(0, xW_1 + b_1\right) W_2 + b2 \tag{3}$$

Nors tiesinės transformacijos skirtingose pozicijose yra vienodos, skirtingiems transformatoriaus sluoksniams naudojami skirtingi parametrai.

1.1.3. Koduotojas ir dekoduotojas

Koduotoją sudaro N vienodų sluoksnių, kur kiekvienas sluoksnis turi du posluoksnius – kelių galvučių savaiminio dėmesio mechanizmą ir pagal padėtį pilnai sujungtą tiesinį tinklą. Kiekvienas iš posluoksnių papildomas liekamojo ryšio ir normalizacijos moduliais.

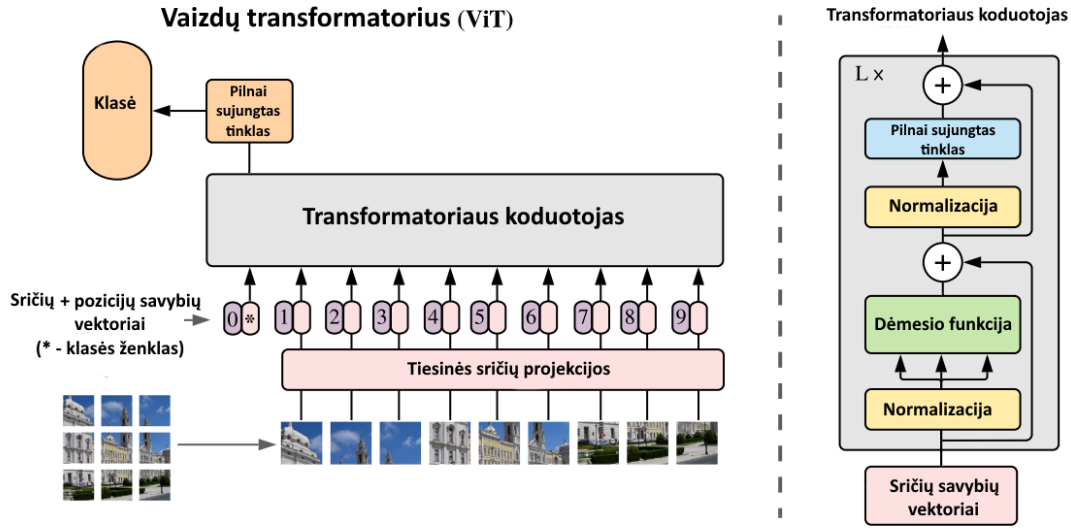
Dekoduotoją taip pat sudaro N vienodų sluoksnių. Prie koduotojo architektūros posluoksnių pridedame papildomą kelių galvučių savaiminio dėmesio mechanizmą, apdorojantį koduotojo išvestį. Kaip ir koduotojo architektūroje, kiekvienas iš posluoksnių papildomas liekamojo ryšio ir normalizacijos moduliais. Taip pat modifikuojame dekoduotojo savaiminio dėmesio posluoksnį (maskavimas), norint išvengti dėmesio skyrimo vėlesnėms išvesties pozicijoms.

1.1.4. Pozicinis kodavimas

Kadangi modelyje nėra pasikartojimo ar konvoliucijos, norint, kad modelis galėtų pasinaudoti sekos eiliškumu, turime įvesti tam tikrą informaciją apie santykinę arba absoliučią ženklų padėtį sekoje. Šiuo tikslu į įvesties savybių vektorius pridedame „pozicinius kodus“. Poziciniai kodai turi tą patį dydį d_{model} kaip ir savybių vektoriai, tad juos galima sumuoti.

1.2. Vaizdų transformatoriai

Po sėkmingų transformatorių modelių pritaikymų natūralios kalbos apdorojimo uždavinių sprendime, nuspręsta pabandyti pritaikyti transformatorius ne tekstinių duomenų apdorojimui, o vaizdinių [DBK+20].



4 pav. Vaizdų transformatorių architektūra [DBK+20]

Vaizdų transformatorių architektūra matome 4 paveikslėlyje. Standartiniam transformatoriui pateikiame vienos dimensijos simbolių savybių vektorių seką. Norint apdoroti dviejų dimensijų vaizdus, turime pertvarkyti kiekvieną vaizdą $x \in \mathbb{R}^{H \times W \times C}$ į suplokštintų dviejų dimensijų sričių seką $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$ (čia H ir W – vaizdo aukštis ir plotis, C – kanalų skaičius, P – kiekvienos vaizdo srities aukštis ir plotis, $N = \frac{HW}{P^2}$ – sričių skaičius). Transformatorius naudoja pastovų išvesties vektoriaus dydį D visuose savo sluoksniuose, todėl suplokštiname sritis ir atvaizduojame jas D dimensijų erdvėje pagal 4 formulę. Šios projekcijos išvestis – vaizdo sričių savybių vektoriai.

$$z_0 = \left[x_{class}; x_p^1 E; x_p^2 E; ...; x_p^N E\right] + E_{pos} \tag{4}$$

Čia $E \in \mathbb{R}^{(P^2 \cdot C) \times D}$ – svorių matrica, $E_{pos} \in \mathbb{R}^{(N+1) \times D}$ – pozicinių kodų matrica.

Kaip ir tekstinių duomenų transformatoriuose, vaizdų sričių savybių vektorių sekos pradžioje pridedame papildomą savybių vektorių ($z_0^0 = x_{class}$), kurio užkoduotą būseną (z_L^0) laikysime vaizdo savybių vektoriumi ($y = LN(z_L^0)$). Šalia savybių vektorių pridedame papildomus pozicinius vektorius, skirtus išlaikyti tą pačią savybių vektorių sekos tvarką tarp visų koduotojo sluoksnių. Tiek išankstinio apmokymo, tiek mokymo metu prie z_L^0 prijungiamas klasifikavimo sluoksnis, kurio išvestis – vaizdo klasė.

Dosovitskiy et al. pastebi, kad vaizdų transformatoriai pasižymi daug mažesniu su specifiniais vaizdais susijusiu indukciniu šališkumu nei konvoliuciniai neuroniniai tinklai. Šios ypatybės priežastis, tai transformatorių savybė, jog savaiminio dėmesio sluoksniai yra globalūs. Papildomai,

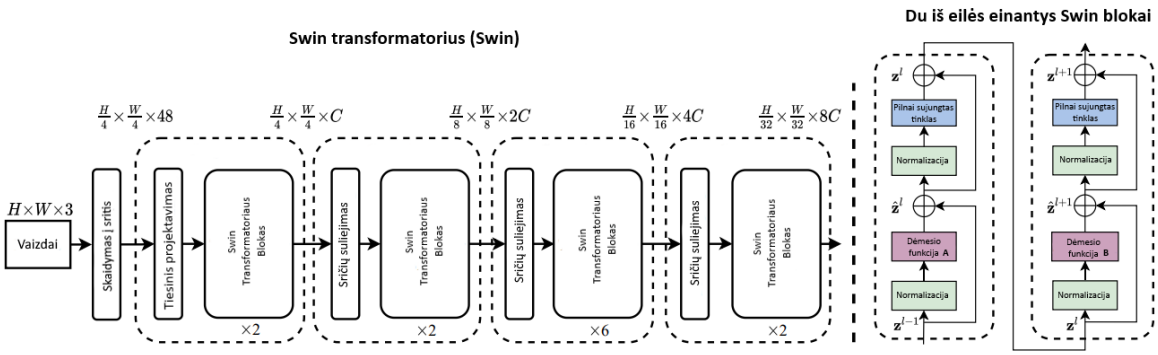
vaizdų sričių padėtis originaliame vaizde nėra naudojama mokymo metu, tad ryšiai tarp atskirų sričių yra atrandami mokymo metu.

1.2.1. Vaizdų transformatorių variantai

Šalia originalaus vaizdų transformatoriaus (ViT) aprašyto Dosovitskiy et al. [DBK+20] geriausiai pasirodančiuose modeliuose galime rasti ir kitą vaizdų transformatorių variantą. Taipogi, visi vaizdų transformatorių modeliai turi dydžio bei kelių kertinių parametrų variacijas.

1.2.1.1. Kelių žingsnių transformatorius

Swin transformatorius – tai 2021 m. pristatytas vaizdų transformatorius, kuriantis hierarchinį vaizdo savybių vektorių ir pasižymintis tiesiniu skaičiavimo sudėtingumu, atsižvelgiant į įvesties vaizdo dydį [LLC+21]. Transformatoriaus architektūra pateikiama 5 paveikslėlyje.



5 pav. Swin vaizdų transformatoriaus architektūra [LLC+21]

Pirmiausia įvesties vaizdas suskirstomas į nepersidengiančias sritis naudojant sričių skaidymo modulį (tokį pat kaip ir ViT). Kiekviena sritis laikoma ženklu, o jos savybių vektorius yra nustatomas kaip neapdorotų pikselių RGB verčių sankaupa. Originaliame Swin transformatoriuje naudojamos 4×4 dydžio sritys, todėl kiekvienos srities savybių vektoriaus dydis yra $4 \times 4 \times 3 = 48$. Šis neapdorotas savybių vektorius yra projektuojamas į pasirenkamą dydį (pav. žymimą C).

Šie vektoriai yra apdorojami keliais transformatorių blokais, kurių savaiminio dėmesio funkcijos yra pakeistos. Transformatorių blokai išlaiko ženklų skaičių $\left(\frac{H}{4} \times \frac{W}{4}\right)$ ir kartu yra laikomi pirmu Swin transformatoriaus žingsniu.

Norint sukurti hierarchinį atvaizdavimą, ženklų kiekis yra mažinamas sujungiant sluoksnius. Pirmasis sričių sujungimo sluoksnis sujungia kiekvienos keturių gretimų sričių grupės požymius ir $4C$ dydžio sujungtus savybių vektorius apdoroja tiesiniu sluoksniu. Taip ženklų skaičius sumažinamas 4 kartus, o išvesties matmuo tampa lygus $2C$. Naujiems savybių vektoriams taikomi Swin transformatoriaus blokai. Pirmasis sričių sujungimo blokas ir požymių transformavimo etapas yra laikomas antru Swin transformatoriaus žingsniu. Po šio žingsnio išvesties rezoliucija yra lygi $\frac{H}{8} \times \frac{W}{8}$.

Procedūra kartojama du kartus, kaip trečias ir ketvirtas Swin transformatoriaus žingsnis, o išvesties resoliucija atitinkamai prilygsta $\frac{H}{16} \times \frac{W}{16}$ ir $\frac{H}{32} \times \frac{W}{32}$. Šie žingsniai kartu sukuria hierarchinį

atvaizdavimą, kurio požymių žemėlapio skiriamoji geba yra tokia pati kaip tipinių konvoliucinių tinklų, pavyzdžiui, ResNet.

1.2.1.2. Modelių dydžiai ir esminiai parametrai

ViT konfigūracijos grindžiamos BERT [DCL+18] naudojamomis konfigūracijomis, kurios apibendrintos 1 lentelėje.

1 lentelė. ViT modelių dydžiai [VSP+17]

Modelis	Sluoksnių skaičius	Išvesties dydis	Daugiasluoksnio perceptrono dydis	Galvučių skaičius	Parametrų kiekis
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Tuo tarpu Swin vaizdų transformatoriai grindžia savo bazinio modelio parametrus ViT-Base modelio parametrais. Taip pat siūlomos Swin-T, Swin-S ir Swin-L, kurių modelio dydžiai atitinka $0.25\times$, $0.5\times$ ir $2\times$ bazinio modelio dydžio.

1.3. Telkimo operacijos

Vietinio telkimo operatoriai, paprastai maksimalus telkimas, yra randami daugelyje konvoliucinių neuroninių tinklų, norint sumažinti priklausomybę nuo mažų pokyčių. Tuo tarpu visuotinis erdvinis sutelkimas paverčia tinklo apskaičiuotą trijų dimensijų aktyvacijų tenzorių į vektorių.

Klasifikacijos užduotyse dažniausiai naudojamas vidurkiu paremtas telkimas, nepriklausantis nuo permutacijų ir teikiantis didesnį visuotinį invariantiškumą pokyčiams. Tuo tarpu vaizdų panašumo vertinimui reikia daugiau lokalizuotos informacijos – tam tikri lokalūs objektai didina vertinimo tikslumą, o globali scenos „netvarka“ (didelis mažų objektų kiekis, netvarkingas objektų išdėstymas) mažina. Dėl to, vaizdų panašumo vertinimui naudojami telkimo operatoriai labiau atsižvelgia į lokalias vaizdo savybes.

1.3.1. Apibendrinto vidurkio telkimo operacija

Apibendrinto vidurkio telkimo (GeM) [RTC18] operacija apskaičiuoja apibendrintą kiekvieno tenzoriaus kanalo vidurkį. Tarkime, jog $x \in \mathbb{R}^{N\times N\times D}$ – tai konvoliucinio neuroninio tinklo kamieno apskaičiuotas savybių vektorius. Pažymime, jog $u \in \Omega = 1,...,N \times 1,...,N$ – tam tikras žemėlapio pikselis, d – kanalas, o x_{du} – tam tikras tenzoriaus elementas. Tuomet GeM sluoksnio savybių vektorius apskaičiuojamas pagal 5 formulę, kur $p > 0$ yra keičiamas parametras.

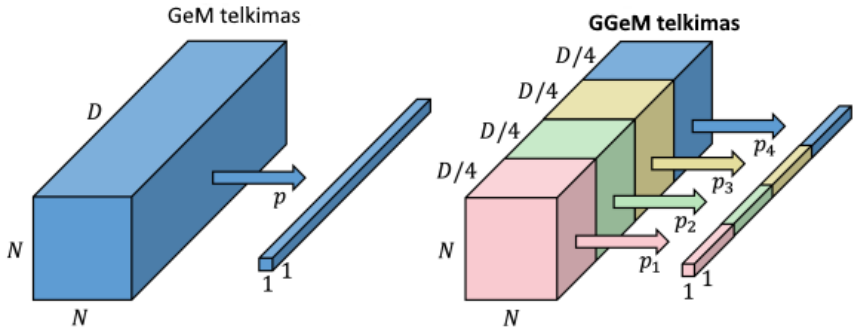
$$\mathbf{e} = \left[\left(\frac{1}{|\Omega|} \sum_{u\in\Omega} x_{du}^{p} \right)^{\frac{1}{p}} \right]_{d=1..D} \tag{5}$$

Nustačius, jog $p > 1$ padidiname sutelktų požymių žemėlapio kontrastą ir sutelkiame dėmesį į svarbiausias vaizdo savybes. GeM – tai vidurkinio telkimo, naudojamo klasifikacijos uždaviniuose ($p = 1$), ir maksimalaus erdvinio telkimo ($p = \infty$) apibendrinimas.

1.3.2. Grupės apibendrinto vidurkio telkimas

Pirmieji vaizdų transformatoriai atitiko įprastą natūralios kalbos apdorojime naudojamų transformatorių architektūrą. Vaizdo savybių vektorius buvo prilyginamas papildomo klasės ženklo paskutinei būsenai. Tačiau naudojant klasės ženklą kaip savybių vektorių iškyla rizika ignoruoti kiekvienos vaizdo srities išvestį ir informaciją. Šiai problemai spręsti vietoje klasės ženklo pradėta naudoti vaizdo sričių išvesčių santalką. ViT atveju naudojamas vidurkio telkimas. Taip pat galima naudoti ir konvoliuciniams tinklams skirtas telkimo strategijas, tačiau jos nėra pritaikytos ViT architektūrai.

Grupės apibendrinto vidurkio telkimas (GGeM) – tai telkimo strategija paremta įvesties kanalų skaidymu į grupes [KKH+22]. Operacijos pavyzdys matomas 6 paveikslėlyje.



6 pav. Telkimo operacijos [KKH+22]

Turint $N \times N \times D$ savybių vektorių, apibendrinto vidurkio telkimo sluoksnis agreguoja vektorius naudodamas tik vieną telkimo parametrą p. Tuo tarpu grupės apibendrinto vidurkio telkimo sluoksnis vykdo grupėmis grįstą agregaciją, su skirtingais telkimo parametrais p_i kiekvienai grupei. Darome prielaidą, kad kiekviena grupė yra nuosekliai išdėstyta išilgai D ašies.

Grupių skaičius G yra laikomas iš anksto nustatytu hiperparametru ir yra prilyginamas vaizdų transformatoriaus galvučių skaičiui. Tuomete $P = \{p_1,...,p_G\}$ – apmokomų parametrų seka. GGeM sluoksnio išvestis – naujas savybių vektorius $e^{(gg)}$, kuris randamas pagal 6 formulę.

$$e_d^{(gg)} = \left(\frac{1}{|\Omega|} \sum_{u \in \Omega_d} x_{du}^{P_{y(d)}} \right)^{\frac{1}{P_{y(d)}}}, \quad y(i) = \left\lceil \frac{i}{D/G} \right\rceil \tag{6.1}$$

$$e^{(gg)} = \left[e_1^{(gg)},...,e_d^{(gg)},...,e_D^{(gg)} \right]^T, \tag{6.2}$$

Jei G prilyginsime 1, GGeM taps apibendrinto vidurkio telkimo sluoksniu. Kita vertus, jei $G = D$, kiekvienas aktyvacijų žemėlapio elementas bus susietas su individualiu parametru.

1.4. Tikslo funkcijos

Vaizdų panašumo vertinimui skirtų neuroninių tinklų mokymui naudojamos įvairios tikslo funkcijos, kurių paskirtis – vertinti ar panašių vaizdų savybių vektoriai yra kuo arčiau vienas kito. Šiame poskyryje apžvelgsime naudojamų tikslo funkcijų ypatumus.

1.4.1. Kryžminės entropijos tikslo funkcija

Atsitiktinio kintamojo X entropija – tai kintamojo galimo rezultato neapibrėžtumo lygis.

$$H(X) = \begin{cases} -\int_x p(x) \log p(x), & \text{jei } X \text{ yra tolydus kintamasis} \\ -\sum_x p(x) \log p(x), & \text{jei } X \text{ yra diskretus kintamasis} \end{cases}. \tag{7}$$

Entropiją apskaičiuojame pagal 7 formulę. Čia X yra atsitiktinis kintamasis, $p(x)$ – tikimybinis pasiskirstymas. Kuo didesnė entropijos H(X) vertė, tuo tikimybių pasiskirstymo neapibrėžtumas bus didesnis, o kuo mažesnė vertė, tuo mažesnis neapibrėžtumas.

Kryžminės entropijos tikslo funkcija – tai tikslo funkcija, skirta klasifikacijos modelių optimizavimui. Klasifikacijos modelio „SoftMax“ sluoksnis paverčia vaizdo savybių vektorių į klasių tikimybes. Kryžminės entropijos tikslas – turint išvesties tikimybes (P), išmatuoti atstumą nuo tiesos reikšmių (T) iki tikimybių. Kryžminės entropijos tikslo vertę skaičiuojame pagal 8 formulę.

$$L_{CE} = -\sum_{i=1}^{n} t_i log(p_i) \tag{8}$$

Čia n – klasių skaičius, $t_i \in T$ – tiesos vertė, $p_i \in P$ – modelio išvesties vertė. Kuo kryžminės entropijos tikslo funkcijos įvertis yra mažesnis, tuo modelio išvesties vertės yra arčiau norimų klasių reikšmių.

1.4.2. Kontrastinė tikslo funkcija

Tarkime, kad turime tris vaizdus I_{A1}, I_{A2} ir I_B. Pirmieji du vaizdai vaizduoja objektą, kuriam priskiriama klasė A, o trečiasis vaizdas – objektą, kuriam priskiriama klasė B. Kontrastinio mokymosi atveju norime sumažinti atstumą tarp panašių pavyzdžių ($\delta(I_{A1}, I_{A2})$) ir padidinti atstumą tarp nepanašių pavyzdžių ($\delta(I_{A1}, I_B)$ ir $\delta(I_{A2}, I_B)$). Čia $\delta(\cdot)$ – atstumo funkcija.

Kontrastinė tikslo funkcija – vienas pirmųjų mokymo tikslų, kuris buvo naudojamas kontrastiniam mokymuisi. Turint panašių arba nepanašių pavyzdžių porą, panašūs pavyzdžiai yra priartinami, o nepanašius – atitolinami. Kontrastinė tikslo funkcija apibrėžta 9 formulėje.

$$L_{constrastive} = (1 - Y)\, \delta(I_i, I_j) + Y max\,(0, \lambda - \delta(I_i, I_j)) \tag{9}$$

Čia I_i ir I_j – vaizdų savybių vektorių pora, Y – žymė, nurodanti ar vaizdai panašūs (0) ar nepanašūs (1), λ – minimalus atstumas tarp nepanašių vaizdų. Detaliau išanalizavus 9 formulę, galima išskirti du skirtingus atvejus:

- Jei vaizdai yra panašūs ($Y = 0$), tuomet mokymo metu minimizuojame narį $\delta(I_i, I_j)$, taip mažindami atstumą tarp panašių vaizdų.
- Kitu atveju ($Y = 1$), mokymo metu minimizuojame narį $max\,(0, \lambda - \delta(I_i, I_j))$, taip didindami atstumą tarp nepanašių vaizdų iki tam tikros ribos λ.

1.4.3. Trejetų tikslo funkcija

Kontrastinės tikslo funkcijos patobulinimas – trejetų tikslo funkcija, pranokstanti pirmąją, nes vietoj porų naudojami trejetai. Trejetų tikslo funkciją apskaičiuojame pagal 10 formulę.

$$L_{triplet} = max\,(0, \delta(I_a, I_p) - \delta(I_a, I_n) + \lambda) \tag{10}$$

Čia I_a – pagrindo vaizdas, I_p – teigiamas paieškos pavyzdys (šio darbo atveju – augmentuotas pagrindo vaizdas), I_n – neigiamas paieškos pavyzdys (šio darbo atveju – augmentuotas kitas vaizdas), o λ – norimas minimalus skirtumas tarp teigiamo ir neigiamo pavyzdžių atstumų.

Sprendžiant tik vaizdų kopijų paieškos užduotį, tokia tikslo funkcija leidžia pasiekti aukštų rezultatų [BJV+19; WSZ+21]. Tačiau sprendžiant vaizdų panašumo vertinimo uždavinį, kuriame ieškome tiek vaizdų kopijų, tiek tai pačiai klasei priklausančių vaizdų, norime, jog vertinimo modelis atsižvelgtų ir į papildomą mokymo metu gaunamą informaciją (pvz. objektų formą, spalvą, klases).

1.4.4. Ketvertų tikslo funkcija

„Sketch-QNet“ – tai ketvertų konvoliucinis tinklas, skirtas sukurti bendrą savybių vektorių erdvę, kurioje galėtume išdėstyti vaizdus, atsižvelgiant į jų spalvą ir formą [FS21]. Modelio tikslas – turint tam tikro objekto spalvotą eskizą, projektuoti vaizdų savybių vektorius taip, kad:

- Tos pačios spalvos ir klasės objektų vaizdų savybių vektoriai būtų arčiausiai eskizo savybių vektoriaus.
- Skirtingos spalvos bet tos pačios klasės objektų vaizdai būtų nutolę nuo eskizo savybių vektoriaus didesniu atstumu nei tos pačios spalvos vaizdai.
- Skirtingos klasės objektų vaizdų savybių vektoriai būtų žymiai nutolę nuo eskizo savybių vektoriaus.

Modelio idėja – ketvertas, kurį aprašome kaip $\Gamma = (I_a, I_{p+}, I_{p+-}, I_n)$. Čia I_a – pagrindo vaizdo savybių vektorius, I_{p+} – tos pačios spalvos ir klasės objekto vaizdo savybių vektorius, I_{p+-} – tos pačios klasės, bet kitokios spalvos objekto vaizdo savybių vektorius, I_n – kitokios klasės objekto vaizdo savybių vektorius. Tuomet, modelio tikslas – projektuoti vaizdus į savybių vektorių erdvę taip, kad $\delta(I_a, I_{p+}) < \delta(I_a, I_{p+-}) < \delta(I_a, I_n)$.

Norėdami patenkinti modelio nelygybę skirstome ją į dvi dalis, kurias matome 11 formulėje.

$$\delta(I_a, I_{p+}) + \alpha \cdot \lambda < \delta(I_a, I_{p+-}) \tag{11.1}$$

$$\delta(I_a, I_{p+-}) + (1 - \alpha) \cdot \lambda < \delta(I_a, I_n) \tag{11.2}$$

Čia λ – norimas minimalus atstumas tarp pozityvaus ir negatyvaus atstumų, o $\alpha \in (0, 1)$ – parametras, reguliuojantis kokiu atstumu I_{p+-} turi būti nutolęs nuo I_{p+} ir I_n. Iš 11 formulės galime išvesti dvi tikslo funkcijas, kurias matome 12 formulėje.

$$l_{triplet1} = max\left(0, \delta(I_a, I_{p+}) + \alpha \cdot \lambda - \delta(I_a, I_{p+-})\right) \tag{12.1}$$

$$l_{triplet2} = max\left(0, \delta(I_a, I_{p+-}) + (1 - \alpha) \cdot \lambda - \delta(I_a, I_n)\right) \tag{12.2}$$

Tikslo funkcijų išvestys galiausiai yra sudedamos, taip gaunant galutinę tikslo funkciją. Metodo autoriai atkreipia dėmesį į siūlomos tikslo funkcijos kitimą mokymo metu. Mokymo pradžioje tiek $l_{triplet1}$, tiek $l_{triplet2}$ yra didesnės už 0. Dėl šios priežasties, mokymo optimizatorius pirmiausia sutelks dėmesį į išraiškos $\delta(I_a, I_{p+}) + \lambda - \delta(I_a, I_n)$ mažinimą. Po to dėmesys skiriamas abiejų 12 formulėje aprašytų tikslo funkcijos dalių mažinimui.

1.4.5. Ribos tikslo funkcija

Kaip minėjome 1.4.2 ir 1.4.3 poskyriuose, kontrastinė tikslo funkcija reikalauja, kad atstumai tarp teigiamų porų būtų mažesni už tam tikrą slenkstinę reikšmę, o atstumai tarp neigiamų porų – didesni, o trejetų tikslo funkcija reikalauja, kad vaizdas būtų arčiau teigiamo pavyzdžio nei neigiamo. Šioms tikslo funkcijoms reikia pritaikyti parametrus, įskaitant tai, kaip imamos vaizdų poros ar trejetai (dažnai sunku nustatyti).

Wu et al. [WMS+17] pasiūlė efektyvų metodą, sprendžiantį šią problemą. Turint vaizdų rinkinį (angl. *batch*), jų savybių vektoriai iš naujo normalizuojami į vienetinį ilgį, atrenkant neigiamas poras kaip funkciją nuo savybių vektorių panašumo. Šios poros tuomet naudojamos ribos (angl. *margin*) tikslo, kontrastinio tikslo varianto, pasižyminčio trejetų tikslo funkcijos privalumais, skaičiavimui.

$$L_{margin}(I_i, I_j, \beta, y_{ij}) = \max\left(0, \alpha + y_{ij}\left(\delta(I_i, I_j) - \beta\right)\right) \tag{13}$$

Ribos tikslo funkcijos apibrėžtį matome 13 formulėje. Čia $y_{ij} \in \{-1, 1\}$ – reikšmė rodanti ar vaizdai sutampa (1) ar nesutampa (-1), $\alpha > 0$ – nekintantis parametras vadinamas riba, $\beta > 0$ – mokymo metu išmokstamas parametras.

$$P_+(B) = \{(I_i, I_j) \in B^2 : Y = 1\}, \tag{14.1}$$

$$P_-(B) = \{(I_i, I_{j^*}) : (I_i, I_j) \in P_+(B), I_{j^*} \sim p(\cdot \mid I_i)\}, \tag{14.2}$$

$$P(B) = P_+(B) \cup P_-(B) \tag{14.3}$$

Tikslas (13) apskaičiuojamas naudojantis teigiamomis ir neigiamomis poromis $(I_i, I_j) \in B^2$, kurių apibrėžimas matomas 14 formulėje.

1.5. Vektorių panašumo vertinimas

Turint vaizdų porą, galime pateikti ją neuroniniam tinklui. Tinklo išvestis – du savybių vektoriai. Jų panašumą galime lyginti keliais skirtingais būdais.

1.5.1. Euklido atstumas

Turint dviejų vaizdų a ir b savybių vektorius $\vec{a} \in \mathbb{R}^N$ ir $\vec{b} \in \mathbb{R}^N$, Euklido atstumas yra atstumo tarp šių vektorių matas. Daugiamatėje erdvėje kiekvienas vektorius yra prilyginamas taškui, o atstumas tarp jų yra lygus tiesės, jungiančios šiuos du taškus, ilgiui.

$$\delta(\vec{a}, \vec{b}) = \sqrt{\sum_{i=1}^{N}(a_i - b_i)^2} \tag{15}$$

Euklido atstumas tarp vektorių $\vec{a}$ ir $\vec{b}$ yra apskaičiuojamas pagal 15 formulę. Euklido atstumas tarp vaizdų savybių vektorių gali būti laikomas vaizdų panašumo įverčiu.

1.5.2. Kosinuso panašumas

Kosinuso panašumas – tai dviejų vektorių panašumo skaičiavimo metodas, kai vektorių skaliarinė sandauga yra dalijama iš kiekvieno vektoriaus dydžių. Kosinuso panašumas parodo kampą tarp vektorių ir yra dažnai naudojamas įvairiuose panašumo vertinimo uždaviniuose (teksto, vaizdų). Šis metodas pasižymi nepriklausymu nuo vaizdų savybių vektorių dydžio ir Euklido atstumo tarp jų. Panašumą galime apskaičiuoti pagal 16 formulę.

$$\cos\Theta(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{||\vec{a}||\,||\vec{b}||} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}} \tag{16}$$

Čia $\vec{a}, \vec{b}$ – dviejų vaizdų a ir b savybių vektoriai. Jei savybių išgavimo metu paversime vektorius vienetinio ilgio vektoriais, kosinuso panašumą galime apskaičiuoti pagal 17 formulę.

$$\cos\Theta(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{||\vec{a}||\,||\vec{b}||} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}} = \frac{\sum_{i=1}^{n} a_i b_i}{1 \times 1} = \sum_{i=1}^{n} a_i b_i \tag{17}$$

Kaip matome, dviejų vaizdų panašumo įvertis yra lygus jų vienetinių savybių vektorių sandaugai. Panašumo įvertis priklauso intervalui nuo -1 iki 1, kur 1 nurodo aukštą panašumą.

Pasinaudoję, 17 formule, galime sukurti dviejų vaizdų aibių panašumo vertinimo metodą. Tarkime, jog $Q = \{\vec{q_1},\vec{q_2},...,\vec{q_n}\}$ – tai užklausos vaizdų savybių vektorių aibė, $R = \{\vec{r_1},\vec{r_2},...,\vec{r_m}\}$ – atitikmens vaizdų savybių vektorių aibė, $\vec{q_i}, 1 \leq i \leq n$ – tam tikro užklausos vaizdo savybių vektorius, $\vec{r_j}, 1 \leq j \leq m$ – tam tikro atitikmens vaizdo savybių vektorius, o n ir m – užklausos ir atitikmens aibių dydžiai. Tuomet panašumo matricą galime apskaičiuoti pagal 18 formulę.

$$S = QR = \begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m1} & s_{m2} & \cdots & s_{mn} \end{pmatrix}, s_{ij} = \sum_{k=1}^{n} q_{ik} r_{jk} = \cos\Theta(\vec{q_i}, \vec{r_j}) \tag{18}$$

Gauname $n \times m$ dydžio panašumo matricą, kurios i-toji eilutė – tai vektorius, kurio $k \in \{1,2,...,m\}$ elementas nurodo panašumą tarp i-tojo užklausos vaizdo ir k-tojo atitikmens vaizdo, o j-tasis stulpelis – vektorius, kurio $l \in \{1,2,...,n\}$ elementas nurodo panašumą tarp j-tojo atitikmens vaizdo ir l-tojo užklausos vaizdo.

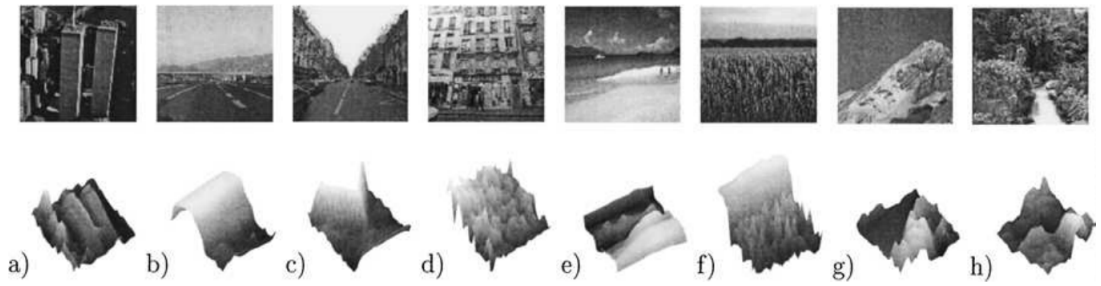
2. Vaizdų panašumo vertinimo metodai

Darbo metu apžvelgti įvairūs egzistuojantys vaizdų panašumo vertinimo metodai – erdvinio apvalkalo metodas, „Multigrain“ neuroninis tinklas, HOW metodas bei „VisionForce“ metodas. Taip pat siūlomas naujas, vaizdų transformatoriais paremtas, metodas. Erdvinio apvalkalo metodas – tai klasikinis vaizdų panašumo vertinimo metodas, o likę – giliaisiais neuroniniais tinklais paremti metodai. Šiame skyriuje plačiau apžvelgsime kiekvieną iš metodų.

2.1. Erdvinio apvalkalo metodas (GIST)

Erdvinio apvalkalo metodas (GIST) – kompiuterinis realaus pasaulio scenų kategorijų atpažinimo modelis, nereikalaujantis atskirų objektų ar regionų segmentacijos ir apdorojimo [OT01]. Metodas paremtas mažų dimensijų savybių vektoriais, vadinamais erdviniais apvalkalais (angl. *spatial envelope*).

Scena paprastai suprantama kaip neapribota objektų konfigūracija, todėl jos semantiniam atpažinimui gali reikėti iš pradžių surasti objektus ir tikslią jų vietą. Scenų pavyzdžiai matomi 7 paveikslėlyje.



7 pav. Įvairios scenos su skirtingais erdviniais apvalkalais ir jų paviršiaus atvaizdais [OT01]

Metodo autoriai siūlo žiūrėti į sceną ne į kaip objektų konfigūraciją, o kaip į atskirą vientisos formos objektą. 7 paveikslėlyje pavaizduoti scenų paviršiaus atvaizdai. Juose 3D užuominos, tikslios formos ar objektai nebėra lengvai pastebimi. Nors scenų vaizdai ir jų paviršiaus atvaizdai saugo tą pačią informaciją, paviršiaus atvaizdus galime laikyti vientisos formos objektais. Kaip ir objektams priskiriamoms kategorijos (pvz. automobilis ar gyvūnas), kurioms priskiriami objektai paprastai atrodo panašūs, scenos priklausančios tai pačiai kategorijai, turi panašią ir stabilią erdvinę struktūrą, kurią galima išskirti neskaidant vaizdo į segmentus.

2.1.1. Erdvinis apvalkalas

Scenos aplinkos erdvinį apvalkalą sudaro sudėtinis ribų, apibrėžiančių aplinkos formą, rinkinys, t.y. ryšiai tarp vaizde esančių paviršių kontūrų ir šių kontūrų savybės.

Norint apibrėžti scenų atpažinimui naudojamo erdvinio apvalkalo savybes, metodo autoriai atliko eksperimentą. Septyniolikos savanorių buvo paprašyta padalyti 81 vaizdą į grupes. Nurodyta, jog toje pačioje grupėje esantys vaizdai privalo turėti panašią globalią struktūrą ar aspektą.

Tiksliai nurodyta, jog negalima naudoti kriterijų susijusių su vaizduose esančiais objektais (pvz. ar yra žmonių, ar ne) ar scenos semantinėmis grupėmis (pvz. miestas, paplūdimys). Užduotis vykdyta trimis etapais – padalyti vaizdus į dvi grupes, padalyti dvi grupes į keturis pogrupius ir padalyti keturis pogrupius į aštuonis pogrupius. Po kiekvieno etapo, savanorių buvo klausiame, kokį kriterijų jie naudojo.

Atsižvelgus į naudotus kriterijus, metodo autoriai pasiūlė naudoti penkias erdvinio apvalkalo savybes:

- Natūralumo (angl. *naturalness*) laipsnį – scenos struktūra žmogaus sukurtoje ir natūralioje aplinkoje smarkiai skiriasi. Žmogaus sukurtos struktūros pasižymi tiesiomis linijomis, o natūrali aplinka – tekstūrų zonomis ir iškiliais kontūrais. Tad scenos pasižyminčios tiesiomis vertikaliomis ir horizontaliomis linijomis turės žemą natūralumo laipsnį, o linijų įvairove pasižymintys vaizdai – aukštą.

- Atvirumo (angl. *openness*) laipsnis – scena gali turėti uždarą erdvinį apvalkalą arba gali tęstis į begalybę (pvz. greitkelis ar jūros krantas). Horizonto linijos egzistavimas nurodo aukštą atvirumo laipsnį, o sceną ribojantys elementai jį mažina.

- Šiurkštumo (angl. *roughness*) laipsnį – scenos savybė, priklausanti nuo scenoje esančių objektų. Šiurkštumas susijęs su scenos fraktaliniais matmenimis ir jos sudėtingumu.

- Plėtros (angl. *expansion*) laipsnį – žmogaus sukurtas struktūras daugiausia sudaro vertikalios ir horizontalios struktūros. Tačiau, atsižvelgiant į stebėtojo požiūrį, statiniai gali būti matomi iš skirtingų perspektyvų. Lygiagrečių linijų susiliejimas suteikia erdvės gylio suvokimą. Plokščias pastato vaizdas pasižymėtų mažu plėtros laipsniu, gatvė su ilgomis nykstančiomis linijomis – dideliu.

- Nelygumo (angl. *ruggedness*) laipsnį – tai žemės paviršiaus nuokrypis horizonto atžvilgiu. Nelygi aplinka sukuria įstrižus kontūrus vaizde ir paslepia horizonto liniją. Dauguma žmogaus sukurtų aplinkų yra ant lygaus paviršiaus, tad tokie vaizdai turi žemą nelygumo laipsnį, o natūralios aplinkos – aukštą.

2.1.2. Erdvinio apvalkalo savybių skaičiavimas

Erdvinio apvalkalo savybių skaičiavimui metodo autoriai pasitelkia du vaizdų atvaizdus – visuotinį energijos spektrą ir spektrogramą. Kiekvieno iš atvaizdų pagrindinės komponentės apibrėžia vaizdais paremtą savybių erdvę, į kurią galima suprojektuoti kiekvieną sceną. Tačiau kiekvienos gaunamos savybės formatas yra sunkiai suprantamas ir nėra reikšmingas stebėtojui be papildomo apdorojimo.

Erdvinio apvalkalo savybės vaizduoja sceną labai mažos dimensijos erdvėje, kurioje kiekvienas matmuo vaizduoja prasmingą scenos erdvės savybę. Jų skaičiavimui pasitelkiami įvairūs regresijos skaičiavimo metodai. Pvz., naudojant tiesinę regresiją, scenos požymis s gali būti apskaičiuojamas pagal 19 formulę.

$$\hat{s} = \mathbf{v}^T \mathbf{d} = \sum_{i=1}^{N_g} v_i d_i = \iint A(f_x, f_y)^2 DST(f_x, f_y) df_x df_y \tag{19}$$

Čia v – scenos vaizdo visuotinio spektro savybės, $A(f_x, f_y)^2$ – vaizdo spektro amplitudė, o $DST(f_x, f_y)$ – diskriminacinis spektrinis šablonas. DST – tai funkcija, nurodanti kaip kiekvienas spektro komponentas prisideda prie erdvinio apvalkalo savybės skaičiavimo [OTG+99; TO99]. Ji pateikiama 20 formulėje.

$$DST(f_x, f_y) = \sum_{i=1}^{N_G} d_i \psi_i(f_x, f_y) \tag{20}$$

Čia d – stulpelinis vektorius, kurio reikšmės pritaikytos GIST mokymo metu, $\psi_i(f_x, f_y)$ – Karhunen-Loeve transformacijos bazinė funkcija, skirta signalo išgavimui.

Erdvinio apvalkalo savybė gali būti apskaičiuota ir naudojantis vaizdo spektrogramos savybėmis.

$$\hat{s} = \mathbf{w}^T \mathbf{d} = \sum_{i=1}^{N_L} w_i d_i = \sum_x \sum_y \iint A(x, y, f_x, f_y)^2 WDST(x, y, f_x, f_y) df_x df_y \tag{21}$$

Lango diskriminacinis spektro šablonas (WDST) – tai funkcija, nurodanti kaip spektrogramos savybės prisideda prie erdvinio apvalkalo savybių skaičiavimo [TO99]. WDST funkcija pateikiama 22 formulėje.

$$WDST(x, y, f_x, f_y) = \sum_{i=1}^{N_L} d_i \psi_i(x, y, f_x, f_y) \tag{22}$$

Kaip ir DST atveju, d – tai stulpelinis vektorius, kurio reikšmės pritaikytos GIST mokymo metu, o $\psi_i(x, y, f_x, f_y)$ – Karhunen-Loeve transformacijos bazinė funkcija. *WDST* reikšmių ženklas, nurodo koreliaciją tarp spektrogramos komponentų ir tam tikros erdvinio apvalkalo savybės.

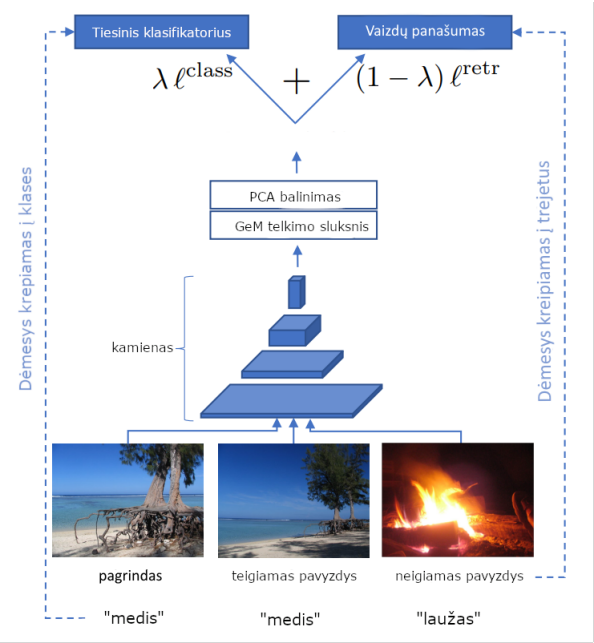
2.1.3. Erdvinio apvalkalo naudojimas

Kiekviena erdvinio apvalkalo savybė atitinka tam tikrą daugiamatės erdvės ašį, kurioje scenos su panašiais erdviniais apvalkalais yra projektuojamos netoli viena kitos. Dėl to, scenos esančios arti viena kitos turėtų būti tos pačios (ar labai panašios) kategorijos, nepriklausomai nuo to, ar erdvinis apvalkalas yra pakankamai prasmingas, kad būtų galima suteikti semantinę scenos kategoriją.

2.2. „MultiGrain“ metodas

MultiGrain – tai vaizdų savybių aprašymo metodas, skirtas spręsti klasifikacijos, objektų paieškos bei panašumo vertinimo problemas (nors šios problemos yra panašios, standartiškai naudojami specializuoti ir nesuderinami vaizdų savybių aprašymo metodai) [BJV+19]. Tai konvoliucinis

neuroninis tinklas, paremtas „ResNet-50“ [HZR+16] architektūra ir papildytas moduliais, leidžiančiais spręsti tiek klasifikacijos, tiek paieškos užduotis. MultiGrain architektūra pateikiama 8 paveikslėlyje.



8 pav. „MultiGrain“ modelio architektūra [BJV+19]

„ResNet-50“ kamienas yra papildomas apibendrinto vidurkio telkimo sluoksniu (aprašytu 1.3.1 skyrelyje), balinimo sluoksniu, naudojamos dvi tikslo funkcijos.

2.2.1. Tikslo funkcijos

Siekiant sujungti klasifikavimo ir paieškos užduotis, metodo autoriai naudoja bendrą tikslo funkciją, kurią sudaro klasifikacijos ir paieškos tikslo funkcijos. Klasifikacijos užduočiai taikoma standartinė kryžminės entropijos tikslo funkcija (aprašyta 1.4.1 skyrelyje), o vaizdų paieškos – ribos tikslo funkcija (aprašyta 1.4.5 skyrelyje). Bendra tikslo funkcija – klasifikacijos ir paieškos tikslo funkcijų derinys, pasvertas koeficientu $\lambda \in [0, 1]$.

$$L = \frac{\lambda}{|B|} \sum_{i \in B} L_{CE}(e_i, \mathbf{w}, y_i) + \frac{1-\lambda}{|P(B)|} \sum_{(i,j) \in P(B)} L_{margin}(e_i, e_j, \beta, y_{ij}) \tag{23}$$

Bendros tikslo funkcijos formulę matome 23 formulėje.

2.2.2. Principinių komponenčių analizės (PCA) balinimas

Norint perkelti savybes, išmoktas naudojantis augmentuotomis standartinėmis vaizdų paieškos duomenų aibėmis, taikomas papildomas PCA balinimo žingsnis. Siekiama, kad Euklido atstumas tarp transformuotų savybių vektorių būtų lygus Mahalanobio atstumui tarp įvesties savybių vektorių. PCA balinimas vykdomas apmokius tinklą, naudojant išorinį nesužymėtų vaizdų duomenų rinkinį. PCA balinimo poveikį galima panaikinti klasifikacijos sluoksnio parametruose,

todėl normalizuoti savybių vektoriai gali būti naudojami ir klasifikacijai, ir vaizdų paieškai. Balinimo operaciją galime užrašyti pagal 24 formulę [GAR+17].

$$\Phi(e) = S\left(\frac{e}{||e||} - \mu\right) \tag{24}$$

Čia S – balinimo matrica, μ – centravimo vektorius, e – vaizdo savybių vektorius. Gauti savybių vektoriai naudojami panašumo vertinimui, naudojantis Euklido atstumu.

2.3. HOW metodas

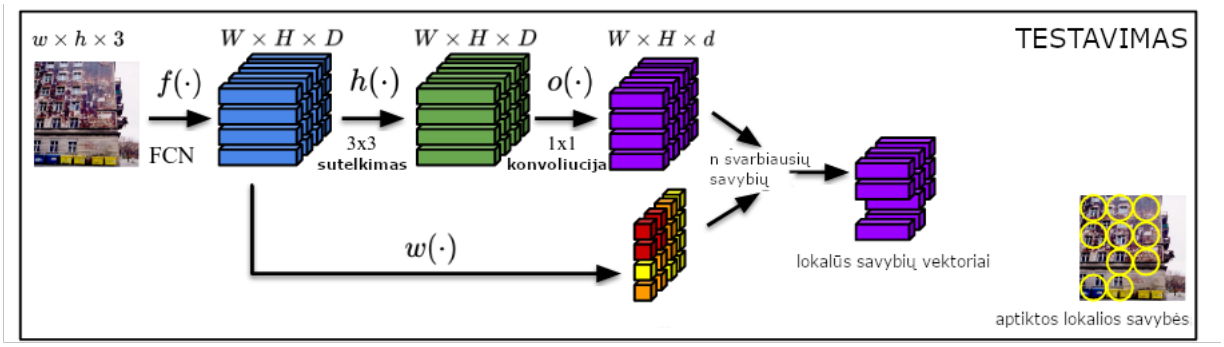
Pavyzdžio lygmens atpažinimo užduotys yra susijusios su labai dideliu klasių skaičiumi ir santykinai nedideliu klasių variantiškumu. Tokių užduočių sprendimui galima naudoti metodus, pagrįstus vietiniais požymiais ir žodžių maišo (angl. *bag of words*) atvaizdu. Tokio metodo pavyzdys – „Video Google“ metodas, skirtas surasti atitinkamą objektą skirtinguose vaizdo įrašo kadruose, naudojantis tekstinės žiniatinklio paieškos metodais [SZ03]. Metodo tikslas – projektuoti vaizdus į savybių vektorius taip, kad panašių vaizdų savybių vektoriai atsidurtų tuose pačiuose klasteriuose, kurie bus laikomi vaizdiniais „žodžiais“ tekstinės paieškos metu.

Noh et al. [NAS+17] pirmieji pasinaudojo globalių savybių vektorių mokymo lankstumu, kad gautų vietinius požymius ir jų savybių vektorius, vadinamus DELF, pavyzdžio lygmens atpažinimo užduoties sprendimui. Vėliau paaiškėjo, kad DELF savybių vektoriai pasiekia geriausius rezultatus, kai yra naudojami kartu su naujausiu vaizdų paieškos metodu, t. y. agreguotu pasirenkamojo atitikimo branduoliu (angl. *Aggregated Selective Match Kernel, ASMK*) [TAJ16].

HOW metodas – lokalių savybių detektorius, paremtas giliuoju neuroniniu tinklu [TJC20]. Architektūra ir tikslo funkcija projektuota taip, kad lokalūs požymiai ir jų savybių vektoriai būtų tinkami naudojimui su ASMK. ASMK pasižymi aukštu našumu net ir nenaudojant erdvinės verifikacijos t. y. tiksli požymių vieta nėra labai svarbi. HOW mokymo metu kelios netoliese esančios vaizdo vietos gali pateikti panašų savybių vektorių – keli panašūs požymiai yra ne slopinami, o išvedamas jų vidurkis.

2.3.1. Metodas

ASMK yra sudėtinga ir daug resursų reikalaujanti operacija (vietinius savybių vektorius turime priskirti vienam iš 65 tūkstančių klasterių). Dėl to HOW metodas pradedamas nuo paprasto karkaso, skirto apskaičiuoti visuotinius savybių vektorius. Tuomet pridedami papildomi moduliai, gerinantys modelio veikimą. Modelis mokomas naudojantis duomenų rinkiniu, kurio vaizdams yra priskirtos žymės, o ASMK naudojamas tik testavimo metu.



9 pav. HOW architektūra [TJC20]

9 paveikslėlyje matome testavimui naudojamą HOW metodo architektūrą. Testavimo metu vaizdų atvaizdavimui yra naudojama n lokalių savybių vektorių. Šie vektoriai ir yra naudojami vaizdų paieškai.

Tarkime, jog $f : \mathbb{R}^{w \times h \times 3} \to \mathbb{R}^{W \times H \times D}$ – pilnai konvoliucinis tinklas, apskaičiuojantis vaizdo I trimatį aktyvacijos tenzorių $f(I)$. Šis tenzorius taip pat gali būti užrašytas kaip kelių dimensijų vietinių savybių vektorių rinkinys $U = \{u \in \mathbb{R}^D\}$, kur D – dimensijų kiekis. Kiekvienas lokalus savybių vektorius yra susietas su raktiniu tašku, dar vadinamu vietiniu požymiu.

2.3.1.1. Vietinis savybių vektorių glaudinimas

Dažnai pasitaiko, jog keliuose aktyvacijų tenzoriaus kanaluose atsiranda didelės aktyvacijos funkcijų vertės, o pačios vertės nėra suderintos. Šios problemos sprendimui, *HOW* metode naudojamas vietinis savybių vektorių glaudinimas, naudojantis vidurkio sutelkimu $M \times M$ kaimynystėje. Rezultatą žymime $\bar{U} = h(f(I))$ arba $\bar{U} = h(U)$, kur h – apmokomų parametrų neturinti telkimo funkcija.

2.3.1.2. Savybių vektorių balinimas

Lokalių savybių vektorių dimensijos yra dekoreliuojamos pasitelkiant balinimo transformaciją. Kartu su balinimu atliekamos dimensijų mažinimo bei vidurkio atimties operacijos. Visos trys operacijos atliekamos, naudojantis funkcija $o : \mathbb{R}^D \to \mathbb{R}^d, o(u) = P(u-m), P \in \mathbb{R}^{d \times D}$. Funkcija $o(\cdot)$ – tai 1×1 konvoliucija su išankstiniu nusistatymu, kur P ir m yra reikšmės, kurias nustatome remiantis mokymo aibės PCA balinimo rezultatais.

2.3.1.3. Naudojamų savybių atranka

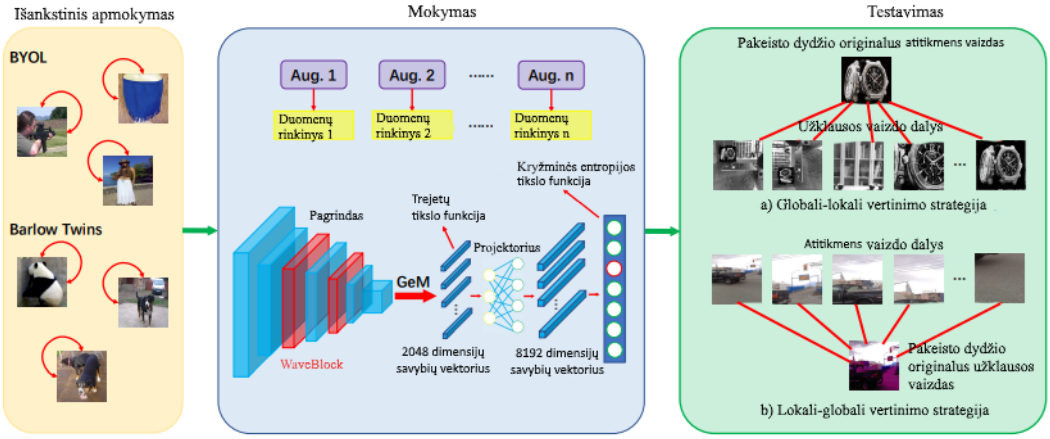
Savybės svarba yra vertinama kaip savybės vektoriaus **u** Euklido norma pagal dėmesio funkciją $w(u) = ||u||$. Mokymo metu savybės svarba naudojama pasverti kiekvieno savybės vektoriaus svarbą atliekant kryžminį lyginimą. Tokiu būdu silpnų požymių poveikis mokymo metu yra ribojamas, o testavimo metu naudojama tik n svarbiausių savybių vektorių.

2.3.2. Testavimo architektūra

Testavimo metu iš pilnai konvoliucinio tinklo gauname n savybių vektorių $o(\bar{u}), u \in U$, kurių svarba, apskaičiuota pagal $w(u)$, yra didžiausia. Šis vektorių rinkinys yra naudojamas ASMK paieškoje, grąžinančioje galutinį metodo rezultatą.

2.4. „VisionForce-mt1“ metodas

„VisionForce-mt1“ (toliau „VisionForce“) – tai duomenimis paremtas lokalios verifikacijos (DPLV) (angl. *data-driven and local-verification, D^2LV*) panašumo vertinimo metodas [WSZ+21].



10 pav. „VisionForce“ metodas [WSZ+21]

Metodą sudaro trys dalys: išankstinio apmokymo, apmokymo bei testavimo.

2.4.1. Neprižiūrimas išankstinis apmokymas

Ne taip, kaip objektų klasifikavimo uždavinyje, kur tokie patys objektai priklauso vienai klasei, vaizdų kopijų paieškos uždavinyje redaguoti vaizdai priklauso tai pačiai klasei kaip ir pradinis vaizdas, o skirtingi vaizdai, vaizduojantys tokį patį objektą – atskiroms klasėms. Šiai problemai spręsti įvairių metodų autoriai pasitelkia naujausius prižiūrimojo mokymosi metodus, apibrėžiančius kiekvieną vaizdą kaip kategoriją ir naudoja duomenų augmentacijų invariantiškumą (angl. *invariance*) pagrindiniam modelio apmokymui [DTP+21].

Tuo tarpu, „VisionForce“ modelio autoriai atmetė save prižiūrinčius mokymosi metodus dėl kelių priežasčių. Pirma, dėl ilgo mokymosi laiko – pvz. norint iš anksto apmokyti „BYOL“ [GSA+20] ar „Momentum^2“ [LLS21] neuroninį tinklą, naudojantis „ImageNet“ duomenų rinkiniu, užtrunkame apie dvi savaites (naudojantis 8 NVIDIA Tesla V100 32GB GPU). Antra, net ir turint neribotus mokymo resursus, metodo autoriams nepavyko išgauti norimo modelio našumo, dėl nepakankamai gerai nustatytų hiperparametrų.

Dėl šių priežasčių, autoriai pasirinko naudoti mokymąsį naudojantis „Momentum^2 Teacher“ [LLS21] (specifinė BYOL versija) bei „Barlow-Twins“ [ZJM+21] metodais. Šiais metodais iš anksto apmokomas modelis, naudojant ImageNet duomenų rinkinį.

2.4.2. Bazinis modelis

Pagrindinio mokymo metu, metodo autoriai remiasi Domainmix [WLZ+20] baziniu modeliu (angl. *baseline*). Sukurtame baziniame modelyje pridedamas GeM telkimo sluoksnis ir WaveBlock [WZL+22] metodas, projektorius į aukštesnes dimensijas bei dvi tikslo funkcijos (kryžminės entropijos ir trejetų). Mokymo metu naudojamas kosinuso atkaitinimas su tiesiniu įsibėgėjimo periodu. Tai mokymosi žingsnio keitimo tipas, kai mokymo pradžioje mokymosi žingsnio daugiklis yra tiesiškai didinamas nuo artimos nuliui reikšmės iki vieneto, o vėliau mažinamas kosinuso formos kreive (tokio proceso formulę matome 25 formulėje). Mokymo žingsnis tuomet yra lygus daugiklio bei pastovaus kintamojo $lr = 0.00035$ sandaugai.

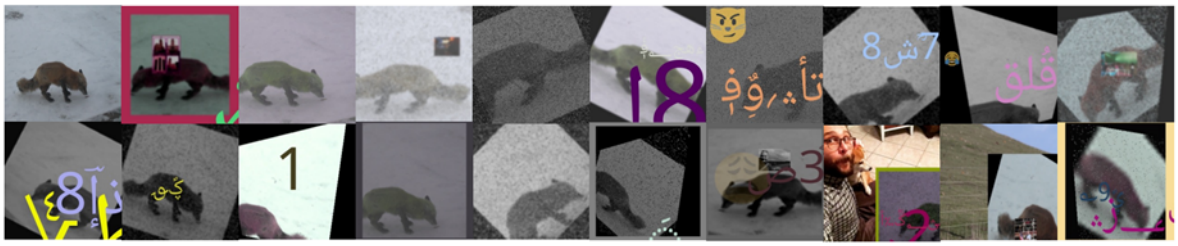
$$daugiklis = \begin{cases} 0{,}99 \cdot \frac{epocha}{5} + 0{,}01, & \text{kai } 0 \leq epocha < 5 \\ 1, & \text{kai } 5 \leq epocha < 10 \\ 0{,}5 \cdot \left(\cos\left(\frac{epocha-10}{15} \cdot \pi\right) + 1\right), & \text{kai } 10 \leq epocha < 25 \end{cases} \tag{25}$$

Modelio mokymui pasirinktos ResNet-50 [HZR+16], ResNet-152 [HZR+16] ir ResNet50-IBN [PLS+18] tinklo pagrindų architektūros. Taip pat naudojamas aukštesnių dimensijų projektorius, projektuojantis 2048 dimensijų savybių vektorių į 8192 dimensijų savybių vektorių. Metodo autoriai, atlikę empirinius bandymus, pastebėjo, jog mokymasis, naudojant didesnių dimensijų vektorius, daro teigiamą įtaką vertinimo modeliams. Modelis mokomas 25 epochas.

Naudojamos dvi tikslo funkcijos: trejetų (naudojama su 2048 dimensijų savybių vektoriumi) bei kryžminės entropijos tikslo funkcija su žymių glotninimu (angl. *label smoothing*) [SVI+16] (naudojama su 8192 dimensijų savybių vektoriumi).

2.4.3. Augmentacijos

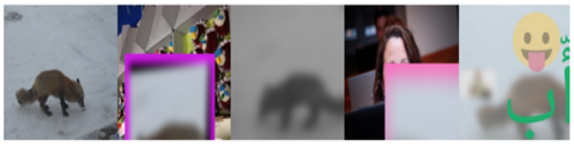
Šis vertinimo metodas yra paremtas duomenimis. Metodo mokymui metodo autoriai sukūrė bazinių vaizdų augmentacijų aibę, kurią papildo šešios sudėtingos augmentacijos.



11 pav. Augmentuotų vaizdų pavyzdžiai. Pirmasis vaizdas – originalas [WSZ+21]

11 paveikslėlyje matome vaizdus, kuriems pritaikyta viena ar kelios bazinės augmentacijos. Kartu šios bazinės augmentacijos sudaro aibę, į kurią įtrauktos tokios klasikinės augmentacijos kaip atsitiktinės pakeisto dydžio iškarpos, atsitiktiniai posūkiai, atsitiktinė pikselizacija, atsitiktinis pikselių sumaišymas, atsitiktinės perspektyvos augmentacijos, atsitiktiniai papildymai (angl. *padding*), atsitiktinis vaizdų pakišimas (angl. *underlay*), atsitiktinis spalvų pakeitimas (angl. *jitter*), atsitiktinis suliejimas, atsitiktinis spalvų panaikinimas (angl. *grayscale*), atsitiktinis horizontalus

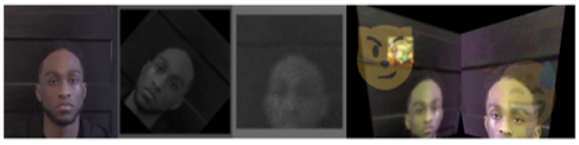
apvertimas (angl. *flipping*), atsitiktinis emoji, teksto ar kitų vaizdų uždėjimas ant vaizdo, vaizdo dydžio keitimas ir t.t.



(a) Super-spalvos augmentacija

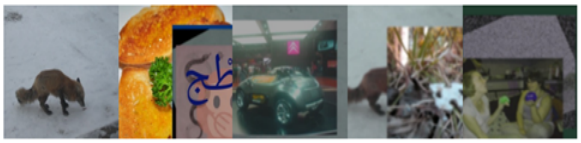
(b) Super-suliejimo augmentacija

(c) Super-tamsos augmentacija



(d) Super-veido augmentacija

(e) Super-nepermatomumo augmentacija



(f) Super-okliuzijos augmentacija

12 pav. Šešios sudėtingos augmentacijos [WSZ+21]

12 paveikslėlyje matome šešias sudėtingas augmentacijas. Kiekviena iš šių augmentacijų pridedama prie bazinės augmentacijų aibės, bet tik po vieną.

Kartu šios augmentacijos sudaro augmentacijų rinkinius, pvz. „bazinė“ (tik bazinės augmentacijos), „bazinė + super-spalvos“, „bazinė + super-okliuzijos“ ir t.t.

Šalia šių augmentacijų naudojama papildoma augmentacija, pakeičianti visus vaizdus į juodai baltus. Ši augmentacija pritaikoma „bazinei“, „bazinei + super-spalvos”, „bazinei + super-suliejimo“ bei „bazinei + super-veido“ augmentacijų aibėms.

Gaunamos 11 skirtingų augmentacijų aibių. Iš anksto apmokytas modelis yra apmokomas su kiekviena iš šių aibių gaunant 11 apmokytų modelių.

2.4.4. Lokali verifikacija

Testavimo metu dėmesys skiriamas dviem ribiniams atvejams:

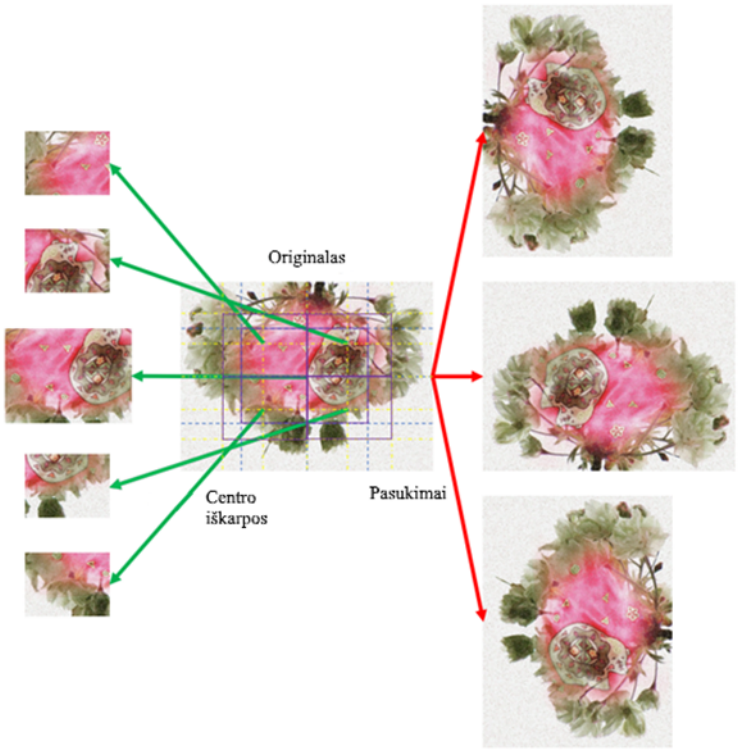
- Kai užklausos vaizdai yra sugeneruojami uždedant atitikmens vaizdą ant nukreipimo vaizdo.
- Kai užklausos vaizdai yra tam tikro atitikmens vaizdo iškarpos.

Modelio autoriai testavimui siūlo naudoti globalią-lokalią bei lokalią-globalią atitikimų paieškos strategiją. Globalios-lokalios strategijos metu tikrinamas globalių atitikmens vaizdų savybių vektorių ir lokalių užklausos vaizdų savybių vektorių atitikimas, o lokalios-globalios – globalių užklausos vaizdų savybių vektorių ir lokalių atitikmens vaizdų savybių vektorių atitikimas.

2.4.4.1. Lokalių užklausos vaizdo savybių vektorių generavimas

Savybių vektorių generavimui metodo autoriai naudoja euristinius ir automatiškai aptiktus aprėpties langelius, taip iškerpant lokalias dalis iš užklausos vaizdų. Lokalių dalių savybių vektoriai yra naudojami kaip užklausos vaizdo lokalūs savybių vektoriai.

Kiekvienam iš užklausos vaizdų dalys yra iš pradžių generuojamos naudojant pasukimą bei centro iškarpas. Vaizdas pasukamas 90, 180 ir 270 laipsnių. Centro iškarpas sudaro tikslaus centro iškarpa bei „1/3 centro" iškarpos. Kartu su originaliu vaizdu, sudaromos 9 dalys, matomos 13 paveikslėlyje.



13 pav. 9 dalys, sugeneruotos sukant ir darant centro iškarpas [WSZ+21]

Naudojantis pasirinktine paieška [UVG+13], generuojami regionų pasiūlymai. Regionų pasiūlymų savybių vektoriai taip pat laikomi lokaliais užklausos vaizdo savybių vektoriais.

Galiausiai, naudojantis YOLOv5 [JSB+20] automatiškai aptinkamos perdangos (angl. *overlay*). Apmokymui metodo autoriai automatiškai sugeneruoja kelis vaizdus su perdangų augmentacijomis bei atitinkamais aprėpties langeliais iš mokymo aibės.

2.4.4.2. Lokalių atitikmens vaizdų savybių vektorių generavimas

Metodo autoriai atitikmens vaizdus suskaido į mažas dalis, skirtas lokaliam atitikimo vertinimui. Kiekvienas atitikmens vaizdas suskaidoma į keturias lygias dalis ir 9 lygias dalis. Tuomet iškerpamas tikslus atitikmens vaizdo centras bei „1/3" centras. Kartu su originaliu vaizdu, metodo autoriai išgauna 19 savybių vektorių iš vieno atitikmens vaizdo.

2.4.5. Modelių kombinacijos

Modelio autoriai pristato du kombinacijų kriterijus: pasitikėjimo bei pilnumo. Tarkime, jog turime vaizdų porą – užklausos vaizdo dalį Q bei originalų atitikmens vaizdą T. Papildomai turime du modelius f ir g. $f(Q,T)$ ir $g(Q,T)$ parodo panašumo įvertį tarp Q ir T, gautus naudojantis modeliais f ir g. Taip pat apibrėžiamos dvi slenkstinės reikšmės α ir β. Naudojant patikimumo kriterijų, panašumo įvertis skaičiuojamas pagal 26 formulę.

$$score_R = \begin{cases} \max\left(f(Q,T), g(Q,T)\right), & \text{jei } f(Q,T) > \alpha \text{ ir } g(Q,T) > \beta \\ \text{Nei vienas}, & \text{kiti.} \end{cases} \tag{26}$$

Įvertis „Nei vienas“ reiškia, kad abu įverčiai yra atmetami, t.y. nėra naudojami galutinio vaizdų poros įverčio skaičiavimui. Naudojant pilnumo kriterijų, įvertis skaičiuojamas pagal 27 formulę.

$$score_F = \max\left(f(Q,T), g(Q,T)\right) \tag{27}$$

Tiek patikimumo, tiek pilnumo kriterijus gali būti papildytas tokiu būdu, kuris palaikytų keletą modelių:

$$score_R = \begin{cases} \max\left(f_1(Q,T), f_2(Q,T), ..., f_n(Q,T)\right), & \text{jei } f_i(Q,T) > \alpha_i \\ \text{Nei vienas}, & \text{kiti.} \end{cases} \tag{28.1}$$

$$score_F = \max\left(f_1(Q,T), f_2(Q,T), ..., f_n(Q,T)\right) \tag{28.2}$$

Čia i priklauso intervalui nuo 1 iki n, $f_1, f_2, ..., f_n$ aprašo n skirtingų modelių, $\alpha_1, \alpha_2, ..., \alpha_n$ aprašo n skirtingų įverčių slenkstinių reikšmių.

Kadangi modelio autoriai naudoja tris skirtingas tinklo pagrindų architektūras, naudojant globalią-lokalią atitikimo strategiją, pasitikėjimo kriterijus naudojamas „ResNet50 + ResNet152“, „ResNet50 + ResNet50-IBN“ ir „ResNet152 + ResNet50-IBN“ kombinacijoms. Naudojant lokalią-globalią atitikimo strategiją, pasitikėjimo kriterijus naudojamas „ResNet50 + ResNet152 + ResNet50-IBN“ kombinacijai. Visoms kitoms kombinacijoms naudojamas pilnumo kriterijus.

Norint kombinuoti skirtingus lokalius-globalius bei globalius-lokalius įverčius, perskaičiuojame pilnumo kriterijų pagal 29 formulę.

$$score_F = \max\begin{pmatrix} \max\left(f(Q_1,T), f(Q_2,T), ..., f(Q_l,T)\right) \\ \max\left(f(Q,T_1), f(Q,T_2), ..., f(Q,T_m)\right) \end{pmatrix} \tag{29}$$

Čia $Q_1, Q_2, ..., Q_l$ aprašo l skirtingų užklausos vaizdo dalių, o $T_1, T_2, \ldots, T_m$ aprašo m skirtingų atitikmens vaizdo dalių.

Suskaičiavus visų modelių ir dalių kombinacijas, galutinis įvertinimas yra laikomas vaizdų poros panašumo įverčiu.

2.5. Klasifikacijos modelio pritaikymas

Pateikus vaizdą klasifikacijos uždaviniui apmokytam konvoliuciniam neuroniniam tinklui, apskaičiuojamas jo savybių vektorius, kuriame išlieka pagrindinės vaizdo savybės. Šį vektorių apdorojus paskutiniu tinklo sluoksniu, vaizdui priskiriama tam tikra klasė. Daugiamatėje erdvėje tam tikrai klasei priklausančių vaizdų savybių vektoriai patenka į grupes, tad pagal juos galima įvertinti vaizdų panašumą. Tai darome pasitelkdami 1.5.2 skyrelyje aprašytą kosinuso panašumą. Vertinimo funkcijai paduodami užklausų ir atitikmens aibių vaizdų savybių vektoriai (priešpaskutinio modelio sluoksnio išvestis) ir sudaroma panašumo matrica. Vaizdai, kurių savybių vektorių kosinuso panašumas yra didžiausias, bus laikomi panašiais vaizdais.

2.6. Vaizdų transformatoriaus pritaikymas

Apmokius vaizdų transformatorių, galime naudoti jo išvestis vaizdų panašumo skaičiavimui. Panašumo vertinimui naudosime 1.5.2 skyrelyje aprašytą kosinuso panašumo metodą.

Transformatoriui pateikus vaizdų rinkinį, koduotojas paverčia $H \times W$ dydžio vaizdus į $\left(\frac{H \times W}{P^2} + 1\right) \times D$ savybių tenzorius. Čia H ir W – vaizdo plotis ir aukštis, P – srities plotis ir aukštis, D – transformatoriaus bloko išvesties dydis. Iš šio tenzoriaus panaikiname pirmą elementą (klasės ženklo vektorių) ir pateikiame likusius vektorius grupės apibendrinto vidurkio telkimo sluoksniui, kuris apibendrina $\frac{H \times W}{P^2} \times D$ dydžio tenzorių į D dydžio galutinį išvesties vektorių. Šiems vektoriams taikome kosinuso panašumo metodą ir sudarome panašumo matricą.

3. Duomenų rinkiniai

Vaizdų panašumo vertinimui dažnai naudojami mašininio mokymosi metodai, kurių veiksmingam mokymuisi reikalingi didelio masto duomenų rinkiniai. Didelis duomenų rinkinys suteikia modeliui daugiau pavyzdžių, todėl išmokstama geriau apibendrinti naujus duomenis, išvengiama persimokymo. Šiame skyriuje apžvelgsime darbo metu naudotus duomenų rinkinius.

3.1. DISC21 duomenų rinkinys

DISC21 duomenų rinkinys – tai rinkinys, naudotas per 2021 m. vaizdų panašumo iššūkį [DTP+21].

Duomenų rinkinį sudaro keturios dalys:

1. Atitikmens vaizdų aibė – milijono vaizdų aibė be jokių augmentacijų.
2. Mokymo vaizdų aibė – milijono vaizdų aibė, surinkta taip pat kaip ir atitikmens vaizdų aibė. Skirta įvairiems mokymo procesams, pvz. įverčių normalizacija, PCA mokymas, modelių mokymas naudojant duomenų augmentacijas.
3. Kūrimo užklausų vaizdų aibė – 10000 vaizdų iš atitikmens vaizdų aibės, sumaišytų su 40000 nukreipimo vaizdų. Kiekvienas iš šių vaizdų buvo redaguotas keliais skirtingais būdais.
4. Testavimo užklausų vaizdų aibė – 10000 vaizdų iš atitikmens vaizdų aibės, sumaišytų su 40000 nukreipimo vaizdų. Kiekvienas iš šių vaizdų buvo redaguotas keliais skirtingais būdais, pasitelkiant skirtingus augmentacijų parametrus ir dar nematytas augmentacijas.

Užklausų aibės vaizdai gali būti dviejų tipų – nukreipimo (angl. *distractor*) ir atitikmens tipo. Nukreipimo užklausų vaizdai – tai augmentuoti vaizdai, kurių originalas nėra įtrauktas į atitikmens aibę, t. y. duomenų bazėje nėra jokio panašaus vaizdo.

Palyginus su realaus pasaulio kasdienėmis užduotimis, duomenų rinkinys yra mažo masto. Rinkinį sudaro du milijonai vaizdų, kas yra tik maža dalis kasdienių pasaulyje apdorojamų vaizdų skaičiaus, o vaizdų augmentacijos yra sudėtingesnės nei augmentacijos, sutinkamos realiose užduotyse.

3.1.1. Duomenų šaltiniai

Atitikmens aibė sudaryta pasitelkiant du šaltinius: YFCC100M duomenų rinkinį [TSF+16] bei DFDC duomenų rinkinį [DBP+20]. Pasirinkti tokie vaizdai, kuriuos būtų galima platinti be teisinių apribojimų. Iš YFCC100M duomenų rinkinio parinkti vaizdai, nevaizduojantys identifikuojamų asmenų (kai vaizdo plotas, vaizduojantis asmenį yra ne didesnis nei 0.5 % viso vaizdo ploto).

Norint atkartoti realias užduotis, į duomenų aibę pridedama vaizdų iš DFDC rinkinio (2020 m. „Facebook“ organizuotas iššūkis). Naudojamos aktorių, davusių leidimą naudoti jų vaizdo įrašus, vaizdai. Maždaug pusė iš vaizdų yra apdoroti, naudojantis „deepfake“ algoritmu, o kiekvienas

iš vaizdų – augmentuotas. Taip panaikinama galimybė atpažinti veidus naudojantis ne panašumo vertinimo, o veidų verifikacijos technikomis.

3.1.2. Išankstinis apdorojimas

DFDC rinkinio vaizdams yra atliekamas vaizdų aptikimas, norint iškirpti juose esančius veidus atmetant fono detales. Tai daroma, norint atkartoti vaizdų tipą, kuris plačiai naudojamas socialiniuose tinkluose – asmenukes (angl. *selfie*).

14 pav. Vaizdų porų, išmetamų iš atitikmens aibės prieš formuojant duomenų rinkinį, pavyzdžiai

Abu duomenų rinkinius sudaro panašių vaizdų poros, kurių panašumas priklauso kelioms skirtingoms kategorijoms (visiškas atitikimas, vaizduojamas tas pats objektas ir kt.). Prieš formuojant duomenų rinkinį, išmetami vaizdai, vaizduojantys tą patį objektą, bet tapatinami su vieno vaizdo dvejomis kopijomis (atmetamų vaizdų pavyzdžiai matomi 14 paveikslėlyje).

3.1.3. Augmentacijos

Vaizdai yra transformuojami naudojantis rankiniu arba automatiniu redagavimu. Kelioms augmentacijoms naudojami papildomi vaizdai iš YFCC100M duomenų rinkinio, t. y. vaizdų uždėjimui ar vaizdo fonui. Užtikrinama, jog papildomi vaizdai nėra įtraukiami į atitikmens aibę.

3.1.3.1. Rankinės augmentacijos

Rankinės augmentacijos buvo vykdomos pasitelkiant išorinius redaguotojus, naudojančius GIMP vaizdų redagavimo priemonę. Redagavimo metu buvo laikomasi tokių taisyklių:

- Pateikiamas vienas pradinis vaizdas ir vienas papildomas vaizdas, kurį galima naudoti koliažų kūrimui.
- Kiekvienas redagavimas turi panaudoti nuo 2 iki 5 skirtingų GIMP įrankių.
- Redagavimo įrankių parametrai turėtų būti keičiami kuo dažniau.
- Vieno vaizdo redagavimas turėtų užtrukti apie 3 minutes.
- Redaguotas vaizdas turėtų būti atpažįstamas.

3.1.3.2. Automatinės augmentacijos

Automatinės augmentacijos buvo vykdomos naudojantis AugLy [PB22] biblioteka ir yra klasifikuojamos į šias kategorijas – uždėjimo (teksto ar emoji), spalvų augmentacijas, pikselių augmentacijas, erdvines augmentacijas ir įterpimo į socialinių tinklų grafinę sąsają augmentacijas.

Kai kuriems vaizdams pritaikyta tik vienos kategorijos augmentacija, kitom – kelios (antra augmentacija taikoma vaizdui su pirma augmentacija). Kurias augmentacijas ir su kokiais parametrais naudoti kiekvienam vaizdui parenkama atsitiktiniu būdu.

3.1.4. AugLy biblioteka

AugLy – tai atviro kodo duomenų augmentacijos biblioteka, teikianti daugiau nei 100 įvairių augmentacijų keturiems duomenų tipams – garsui, vaizdams, tekstui ir vaizdų įrašams [PB22]. Biblioteka atkartoja veiksmus, kurios vartotojai atlieka su duomenimis kiekvieną dieną. Teikiamos funkcijomis paremto ir klasėmis paremto formato augmentacijas bei intensyvumo funkcijos, leidžiančios nustatyti augmentacijos stiprumą pagal tam tikrus parametrus.

Taip pat teikiami operatoriai, leidžiantys apjungti skirtingas augmentacijas, taikyti augmentacijas su tam tikra tikimybe bei taikyti kelias kategorijas (pvz. augmentuojant tiek vaizdo įrašo garso takelį, tiek kiekvieną vaizdą).

Skirtingai nuo kitų vaizdų augmentavimo bibliotekų, AugLy susitelkia į tokias augmentacijas, kokias vartotojai atlieka internete, pvz. teksto ar *emoji* uždėjimas ant vaizdo (šios augmentacijos nėra kitose „AugLy“ bibliotekos autorių apžvelgtuose bibliotekose).

3.2. Google Landmarks V2 duomenų rinkinys

Google Landmark Retrieval iššūkis – tai 2021 m. Google suorganizuotas renginys[1], kurio, dalyvių buvo prašoma sukurti metodus panašių vaizdų aptikimui didelėje duomenų bazėje. Užklausos vaizdas – tai tam tikro žymaus pasaulio objekto (angl. landmark) vaizdas, o užduotis – atrasti visus šio objekto vaizdus duomenų bazėje. Iššūkis pritraukė 541 komandą. Metodų vertinimui pasitelktas mikro vidutinis preciziškumas, o geriausiai pasirodžiusiems dalyviams įteikti piniginiai prizai.

Iššūkiui naudotas Google Landmarks Dataset v2 (GLDv2) [WAC+20] duomenų rinkinys, kurio tikslas – atkartoti industrinio lygio žymių objektų atpažinimo sistemų sprendžiamas užduotis. Duomenų rinkinys pasižymi šiomis savybėmis:

- Didelio mąsto – norint apimti viso pasaulio objektus, rinkinį privalo sudaryti milijonai vaizdų.
- Kintamumas klasėje – vaizdai daromi su skirtingomis apšvietimo sąlygomis bei pozicijomis. Į rinkinį taip pat įtraukiami vaizdai, kurie nevaizduoja pačio objekto, tačiau yra su juo susiję (pvz. vaizdai padaryti ant objekto, objektų piešiniai ir k.t.).

[1] `https://www.kaggle.com/c/landmark-retrieval-2021`

- Klasių disbalansas (angl. Long-tailed class distribution) – žymių objektų (pvz. Eifelio bokšto) vaizdų yra kur kas daugiau nei mažiau žinomų (pvz. Klaipėdos arkos).
- Užklausos už dalykinės srities ribų – užklausų srautas, kurį gauna šios sistemos, gaunamas iš įvairių programėlių. Dalis, kurią sudaro žymių objektų vaizdai, yra tik maža dalis visų vaizdų (vaizduojančių kitas objektų kategorijas). Keliama atpažinimo algoritmo atsparumo problema.
- Atviros licencijos vaizdai – naudoti tik vaizdai, kurių licencijos leidžia juos laikyti neribotą laiką ir pernaudoti publikacijose.

3.2.1. Google Landmarks V2 savybės

Duomenų rinkinį sudaro virš 5 milijonų vaizdų, sužymėtų daugiau nei 200 tūkstančių skirtingų žymių. Duomenų rinkinys skirstomas į tris rinkinius:

1. 118 tūkstančių užklausų vaizdų su pirminės tiesos anotacijomis.
2. 4,1 milijono mokymo vaizdų (vaizduojami 203 tūkstančiai žymių objektų) su žymėmis, kurias galima naudoti vaizdų atpažinimo modelių mokymui.
3. 762 tūkstančius atitikmens vaizdų, kuriuos galima naudoti vaizdų panašumo vertinimo modelių mokymui.

Taip pat teikiamas mažesnis mokymo rinkinys, sudarytas iš 1,2 milijono vaizdų (vaizduojami 15 tūkstančių objektų).

Duomenų rinkinys apima žymius objektus iš 246 šalių iš ISO 3166-1 šalių sąrašo (iš viso 249). Nors rinkinys apima visą pasaulį, jo negalima laikyti reprezentaciniu – kiekvienos šalies vaizdų kiekis priklauso nuo tos šalies Wikimedia Commons bendruomenės. Maždaug 28% vaizdų vaizduoja natūralius objektus (pvz. ežerai, salos, upės), o likę 72% – žmogaus pagamintus objektus (pvz. bažnyčios, muziejai, tiltai).

3.2.2. Duomenų šaltiniai

Pagrindinis duomenų šaltinis – Wikimedia Commons[2]. Tai vaizdų registras, kuriame laikomi keli milijonai žymių objektų vaizdų, teikiamų pagal Creative Commons ar Public Domain licencijas. Didžioji dalis vaizdų surenkama per kasmetinį „Wiki Loves Monuments“ iššūkį, kurio tikslas sukelti kuo daugiau aukštos kokybės, laisvai licencijuojamų žymių objektų vaizdų ir juos sužymėti. Kartu su Wikimedia Commons, vaizdai rinkti pasitelkiant savanorius operatorius, kurie buvo išsiųsti nufotografuoti tam tikrus žymius objektus naudojantis išmaniaisiais telefonais.

[2] `https://commons.wikimedia.org/wiki/Main_Page`

4. Eksperimentas

Darbo metu vykdytas naujo vaizdų transformatoriais paremto vaizdų panašumo vertinimo modelio mokymas, egzistuojančių bei apmokyto modelio testavimas. Šiame skyriuje apžvelgsime eksperimento vykdymo aplinką. mokymo ypatumus, testavimo metodiką ir rezultatus.

4.1. Mokymo ir testavimo aplinka

Eksperimentas vykdytas VU Informacinių technologijų paslaugų centro paskirstytų skaičiavimų tinkle. Tiek mokymosi tiek testavimo metu naudotasi virtualiomis mašinomis su tokiomis komponentėmis: Intel© Xeon© E5-2698 v4 procesoriumi (5 branduoliai), NVIDIA© Tesla V100-SXM2-32GB grafiniu procesoriumi ir 70 gigabaitų operatyviosios atminties. Tiek mokymo tiek testavimo kodas įgyvendintas Python programavimo kalba (3.8 versija) ir šiais pagrindiniais paketais: pytorch, numpy, Pillow, datasets, transformers. Rezultatų vizualizacijoms naudotas matplotlib paketas.

4.1.1. Duomenų rinkiniai

Vaizdų panašumo vertinimo metodų mokymui ir testavimui pasirinkta naudoti DISC21 ir Google Landmarks V2 duomenų rinkinių poaibius.

4.1.1.1. DISC21 poaibis

Kaip minėta 3.1 poskyryje, DISC21 duomenų rinkinį sudaro milijono atitikmens vaizdų aibė, milijono mokymo vaizdų aibė bei po 50000 vaizdų turinčios kūrimo bei testavimo užklausų vaizdų aibės. Užklausų aibės susideda iš 10000 atitikmens vaizdų sumaišytų su 40000 nukreipimo vaizdų.

Sudarome mažesnę mokymo duomenų aibę $T_{DISC21} = \{t_0, t_1, ..., t_{99999}\}$. Aibė sudaroma atsitiktinai parenkant 100000 (10 %) vaizdų iš pilnos mokymo aibės.

Eksperimentui pasirinkti pirmi 10000 testavimui skirti užklausos vaizdai. Pažymėkime, kad $Q_{DISC21} = \{q_0, q_1, ..., q_{9999}\}$. Gautame poaibyje 1989 vaizdai turi atitikmens vaizdą, o likę 8011 yra nukreipimo vaizdai.

Kiekvienam iš 1989 atitikmenį turinčių užklausos vaizdų parenkame jo atitikmenį bei atitikmens aplinkoje esančius vaizdus (bendras lango ilgis – 50). Šiuo metodu gauname $1989 \times 50 = 99450$ atitikmens vaizdų, kuriuos papildome 550 atsitiktiniais atitikmens vaizdais, kurie dar nebuvo atrinkti. Gauta atitikmens vaizdų aibė $R_{DISC21} = \{r_0, r_1, ..., r_{99999}\}$.

4.1.1.2. Google Landmarks V2 poaibis

3.2 poskyryje aprašytas Google Landmarks V2 duomenų rinkinys, kurį sudaro 118 tūkstančių užklausų vaizdų, 4,1 milijono mokymo vaizdų ir 762 tūkstančiai atitikmens vaizdų.

Google Landmarks V2 mokymo vaizdai yra sužymėti viena iš 81313 klasių (tam tikro žymaus objekto kodas). Sudarydami mokymo aibę iš pradžių atrenkame klases, kurioms yra priskirta daugiau nei 20 vaizdų. Po šio filtro pritaikymo lieka 19693 žymių objektų. Iš šių objektų atsitiktinai

parenkame 5000 objektų. Tuomet, jei objektas yra siejamas su ne daugiau nei 50 vaizdų, naudojami visi to objekto vaizdai. Kitu atveju naudojame 20 atsitiktinai parinktų to objekto vaizdų. Taip sudarome mažesnę mokymo duomenų aibę $T_{LAND} = \{t_0, t_1, ..., t_{138519}\}$.

Šiame duomenų rinkinyje 1129 užklausų vaizdai turi vieną ar kelis atitikmens vaizdus, likę vaizdai yra nukreipimo. Šie 1129 vaizdai yra siejami su 3081 skirtingu atitikmens vaizdu. Kaip matome tik maža dalis atitikmens vaizdų yra susieti su testavimo aibės užklausų vaizdais.

Sudarydami Google Landmarks V2 poaibius naudosime visas užklausas, turinčias bent po vieną atitikmens vaizdą, o šią aibę pažymėsime $Q_{LAND} = \{q_0, q_1, ..., q_{1128}\}$. Tuomet atrenkame atitikmens vaizdus, kurie yra priskirti bent vienam užklausų vaizdui. Gautą aibę papildome 6919 atsitiktinai parinktais atitikmens vaizdais, sudarant aibę $R_{LAND} = \{r_0, r_1, ..., r_{9999}\}$.

4.1.2. Vertinimo metrikos

Tyrimo metu buvo atsižvelgiama į keletą metrikų:

- Vaizdų apdorojimo greitis
- Vaizdų lyginimo greitis
- Metodo preciziškumo ir atkūrimo metrikos

Vaizdų apdorojimo ir lyginimo greičiai leidžia įvertinti metodo panaudojamumą. Kuo greičiau galime apdoroti naujus duomenis, tuo greičiau galime pateikti panašumo įvertį būsimam metodo vartotojui. Metodų rezultatų vertinimui naudosime mikro vidutinį preciziškumą ir atkūrimo metrikas.

4.1.3. Rezultatų vertinimas

Vaizdų panašumo vertinimo metodo išvestis – tai porų rinkinys (kiekvieną porą sudaro užklausos vaizdas ir kandidatinis atitikmens vaizdas) ir kiekvienos poros panašumo įvertis.

Algoritmo aptiktų porų preciziškumas ir atkūrimas vertinami lyginant juos su etaloniniais žymenimis. Pasirinkus tam tikrą poros panašumo slenkstinę reikšmę (naudojame tik N aukščiausius panašumo įverčius turinčių porų), atrenkame poras, kurias laikysime aptiktomis poromis. Laikysime, kad aptikta pora yra neteisinga, jei:

- Aptiktas užklausos vaizdas yra nukreipimo vaizdas, t. y. šis vaizdas neturi atitikmens.
- Aptikto atitikmens atvaizdo žymė nesutampa su užklausos vaizdo žyme.

Keičiant slenkstinę porų įtraukimo reikšmę, galime valdyti preciziškumo ir atkūrimo metrikų santykį. Šias metrikas galime pritaikyti ir didesniems duomenų rinkiniams, net jei tuose rinkiniuose nėra papildomų vaizdų porų. Pvz., jei naudojant 100000 atitikmens vaizdų aibę, metodo preciziškumas yra lygus 75 %, o atkūrimas lygus 60 %, tuomet, praplėtus atitikmens aibę iki vieno milijono vaizdų, atkūrimo metrika liks nepakitusi (nėra papildomų teigiamų porų), o preciziškumas kris iki

23 % (10 kartų padidėja klaidingai teigiamų porų kiekis). Dėl šio preciziškumo sumažėjimo, apdorojant didelius duomenų rinkinius, vaizdo panašumo vertinimo algoritmai dažnai siekia tiekti aukštą preciziškumą ir žemą atkūrimą [DTP+21].

Norėdami vertinti metodo rezultatus visų galimų slenkstinių reikšmių atveju, naudojame metriką, vadinamą mikro vidutiniu preciziškumu (µAP). Ši metrika yra lygi plotui po preciziškumo atkūrimo kreive (kai vertinimui naudojamos visos poros) ir yra apskaičiuojama pagal 30 formulę.

$$\mu AP = \sum_{i=1}^{N} p(i)\Delta r(i) \in [0, 1] \tag{30}$$

µAP skaičiuojamas sumuojant preciziškumą kiekvienoje surūšiuoto porų sąrašo pozicijoje, pasvertą pagal tos pozicijos ir ankstesnės pozicijos atkūrimo skirtumą. Gauta reikšmė išreiškiama procentais, didesnė reikšmė nurodo geresnį vertinimo rezultatą.

Rezultatų vertinimo metrikas žymėsime tokiu būdu:

- μAP – mikro vidutinis preciziškumas;
- R@P=90 – atkūrimas, kai preciziškumas lygus 90 %. Jei toks preciziškumas nepasiekiamas, žymėsime įvertį „–“ ženklu;
- R@Rank1 – atkūrimas, kai vertinimui naudojama tik aukščiausią įvertį turinti vaizdų pora;
- R@Rank10 – atkūrimas, kai vertinimui naudojama 10 aukščiausius įverčius turinčių vaizdų porų.

4.2. Mokymo procesas

Naudojantis 1 skyriuje aprašytomis architektūromis ir operacijomis buvo sukurtas naujas vaizdų panašumo vertinimo modelis. Šiame poskyryje apžvelgsime mokymo parametrus, tikslo funkcijų naudojimą, mokymo rezultatus.

4.2.1. Vaizdų transformatorių variantai

Norint parinkti tolimesniame mokyme naudojamą vaizdų transformatoriaus variantą buvo atlikta kelių transformatorių variantų analizė. Analizė vykdyta naudojantis Q_{DISC21} ir R_{DISC21} duomenų poaibiais. Analizės rezultatai pateikiami 2 lentelėje.

Čia ViT – originalaus transformatoriaus [DBK+20] variantai, Swin, Swin2 ir EsViT – kelių žingsnių transformatorių variantai. Variantai, kurie yra pažymėti „(22)“ ženklu buvo iš anksto apmokyti naudojantis „ImageNet-21k“ duomenų rinkiniu, likę – „ImageNet-1k“ duomenų rinkiniu. Kaip matome, geriausią rezultatą parodė „ViT-L“ vaizdų transformatoriaus variantas. Tolimesniuose poskyriuose taikysime tokią notaciją:

- ViT-D – ViT-L vaizdų transformatorius, apmokytas naudojantis DISC21 duomenų rinkiniu ir trejetų tikslo funkcija;

2 lentelė. Vaizdų transformatorių variantų vertinimo rezultatai

Variantas	μAP	R@P=90	R@Rank1	R@Rank10
EsViT-B	0,00039	–	0.01408	0.01810
EsViT-T	0.00042	–	0.01307	0.01559
Swin-B (22)	0.03161	0.01911	0.17446	0.19658
Swin-T	0.01888	0.01156	0.09402	0.11413
Swin2-B	0.00552	–	0.12770	0.15435
Swin2-T	0.02454	0.01357	0.10256	0.12117
ViT-B	0.04938	–	0.25339	0.30468
ViT-B (22)	0.04228	0.01508	0.24585	0.29965
ViT-L	**0.05027**	**0.01810**	**0.26496**	**0.31574**
ViT-L (22)	0.03925	0.01207	0.23529	0.29261

- ViT-LT – ViT-L vaizdų transformatorius, apmokytas naudojantis LAND duomenų rinkiniu ir trejetų tikslo funkcija;
- ViT-LQ – ViT-L vaizdų transformatorius, apmokytas naudojantis LAND duomenų rinkiniu ir ketvertų tikslo funkcija.

4.2.2. Vaizdų augmentacijos

Visų modelių mokymo metu naudotos Wang et al. [WSZ+21] pristatytas bazinių augmentacijų rinkinys. Augmentacijas taikomos realaus laiko principu, t. y. vaizdai augmentuojami ne vieną kartą mokymo pradžioje (išankstinio pasiruošimo principas), o kiekvienos iteracijos metu (su tam tikra tikimybe).

4.2.3. Naudotos tikslo funkcijos

ViT-D ir ViT-LT mokymo metu naudota standartinė trejetų tikslo funkcija. Tarkime, jog turime vaizdą I_a. Tai pagrindo vaizdas. Tuomet vaizdo I_a augmentuotą vaizdą I_{aa} laikysime teigiamu pavyzdžiu I_p. Neigiamu pavyzdžiu I_n laikysime kito vaizdo $I_b \neq I_a$ augmentuotą vaizdą I_{ab}.

ViT-LQ mokymo metu naudojame ketvertų tikslo funkcija. Kaip ir kitų modelių atveju turime pagrindo vaizdą I_a. Tuomet vaizdo I_a augmentuotą vaizdą I_{aa} laikysime teigiamu pavyzdžiu I_{p+}. Vaizdo $I_c \neq I_a$, žyminčio tą patį objektą, augmentuotą vaizdą I_{ac} laikysime pusiau teigiamu pavyzdžiu I_{p+-}. Neigiamu pavyzdžiu I_n laikysime kito vaizdo $I_b \neq I_a \neq I_c$, žyminčio kitą objektą nei I_a, augmentuotą vaizdą I_{ab}.

Kadangi DISC duomenų rinkinyje nėra sužymėtų klasių, mokant vaizdų transformatorių, naudojantis DISC duomenų rinkiniu, ketvertų tikslo funkcijos naudoti negalime.

4.2.3.1. Itin sunkių pavyzdžių paieška

Atsižvelgus į trejetų ir ketvertų tikslo funkcijų apibrėžimus nesunku suprasti, jog parenkant neigiamą pavyzdį I_n atsitiktinai, dažniausiai patenkinsime tikslo funkcijų nelygybes. Dėl šios priežasties, modelis nebesimokys, t. y. tikslo funkcijų įverčiai išliks tokie patys ar panašūs. Šios

problemos sprendimui, nuspręsta naudoti itin sunkių pavyzdžių paiešką. Tai trejetų ir ketvertų tikslo funkcijų optimizavimo metodas, kai vietoje vieno neigiamo pavyzdžio atsitiktinai parenkami $n = 24$ neigiami pavyzdžiai (visi tenkina neigiamo pavyzdžio apibrėžimus).

Turint n neigiamų pavyzdžių, paverčiame kiekvieną iš jų į savybių vektorių. Tuomet skaičiuojame atstumą nuo pagrindo vaizdo savybių vektoriaus iki galimo neigiamo pavyzdžio savybių vektoriaus. Mažiausiu atstumu nuo pagrindo vaizdo nutolęs neigiamas pavyzdys yra naudojamas tikslo funkcijos skaičiavimo metu.

4.2.4. GGeM naudojimas

Visų modelių mokymo metu naudotas 1.3.2 aprašytas grupės apibendrinto vidurkio telkimo sluoksnis. Jis naudojamas pagal 2.6 aprašytą vertinimo metodą, tik vietoj panašumo matricos sudarymo, telkimo sluoksnio išvestis pateikiama tikslo funkcijoms. Naudojama 16 grupių.

4.2.5. Mokymo rezultatai

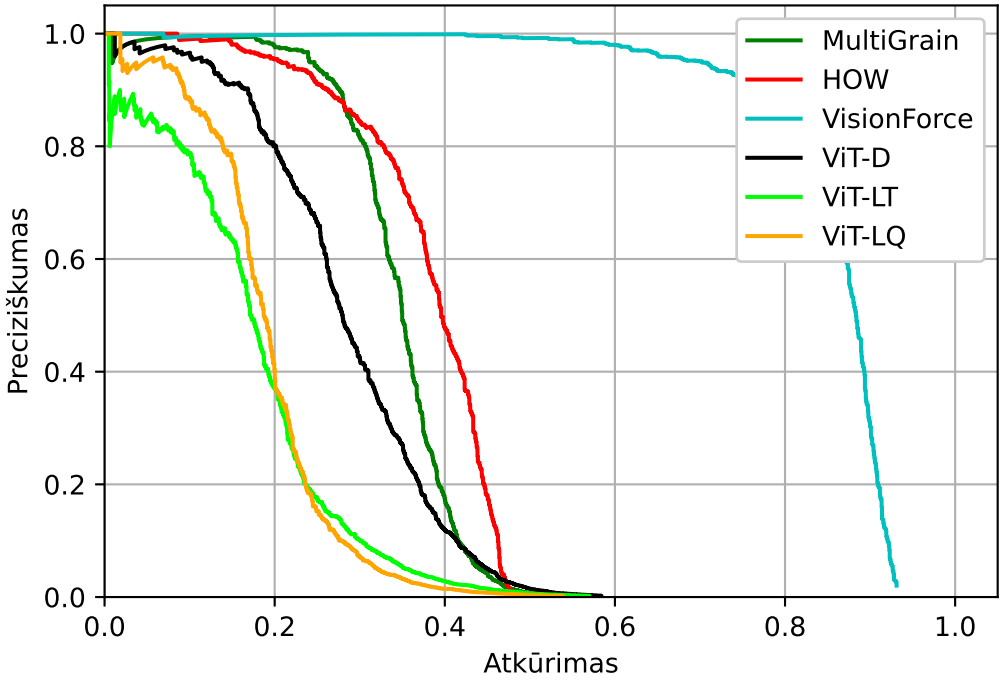
Kiekvienas iš modelių buvo mokytas 25 epochas, kurių metu buvo apdorojami visi mokymo aibių vaizdai. Tikslo funkcijų įverčių kitimo mokymo metu grafikai yra pateikiami priede nr. 1. Pažymima, jog pradedant 11 mokymo epocha, naudojama itin sunkių atvejų paieška. Šis mokymo metodo pakeitimas padidina tikslo funkcijų įverčius visų trijų modelių mokymo metu.

4.3. Testavimo procesas

Testavimas vykdytas apdorojant užklausų ir atitikmens aibių vaizdus vaizdų panašumo vertinimo modeliai ir sudarant panašumo matricą, naudojant kosinuso panašumą arba Euklido atstumą kaip panašumo įvertį.

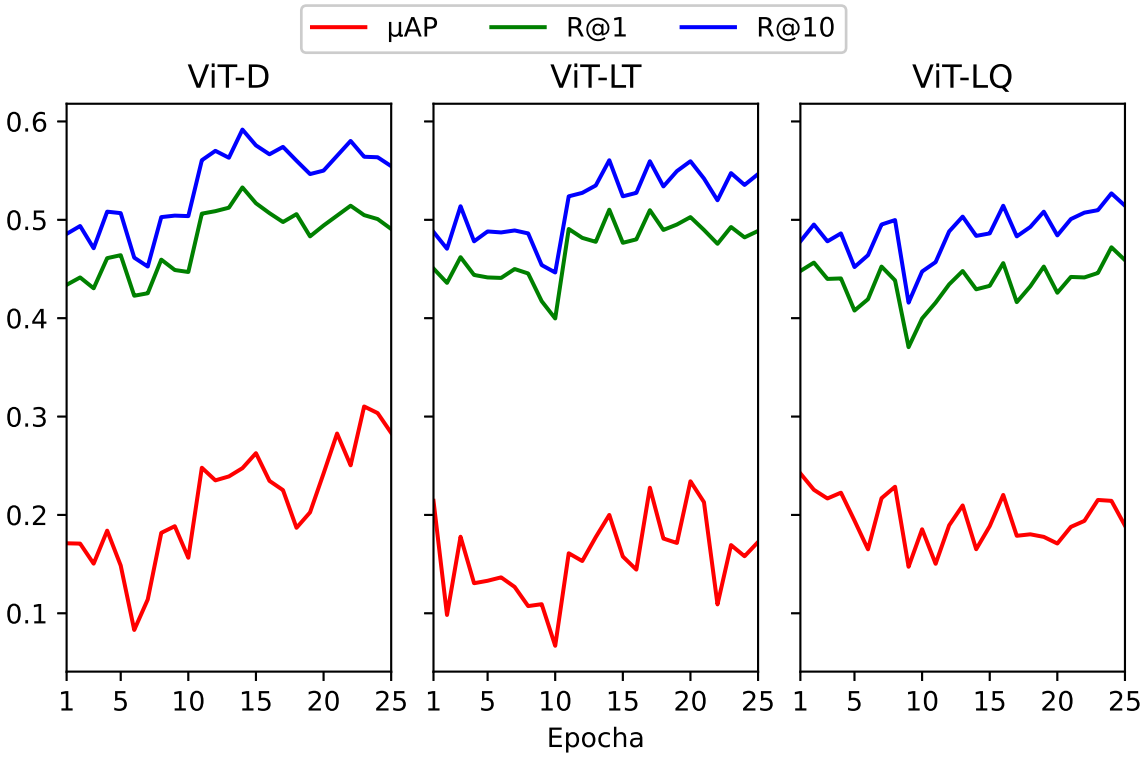
3 lentelė. Metodų vertinimo rezultatai, naudojant DISC21 duomenų rinkinį

Metodas	µAP	R@P=90	R@Rank1	R@Rank10
GIST	0,088	0,064	0,183	0,203
MultiGrain	0,346	0,277	0,423	0,496
HOW	0,380	0,263	0,462	0,488
VisionForce	**0,858**	**0,775**	**0,909**	**0,930**
Klasifikacijos modelio pritaikymas (ResNetV2-101)	0,028	0,018	0,234	0,276
Klasifikacijos modelio pritaikymas (InceptionV3)	0,035	0,019	0,213	0,258
ViT-D	0,283	0,169	0.491	0,555
ViT-LT	0,172	0,018	0,489	0,547
ViT-LQ	0,189	0,094	0,459	0,514



15 pav. Metodų preciziškumo atkūrimo kreivės, naudojant DISC21 duomenų rinkinį

Iš duomenų, pateiktų 3 lentelėje ir 15 paveikslėlyje (pateikiami šeši geriausius rezultatus parodę metodai), matome, kad geriausius rezultatus pasiekia „VisionForce“ metodas (preciziškumas 0,858). Antroje vietoje – HOW metodas. Klasifikacijos modelio pritaikymo metodas yra paskutinėje vietoje. To buvo galima tikėtis, nes metodas remiasi klasifikacijos uždaviniams skirtais neuroniniais tinklais, kurie nėra pritaikyti vaizdų panašumo užduočiai. Vaizdų transformatoriais paremti modeliai atsilieka nuo HOW, „VisionForce“ ir MultiGrain metodų µAP atžvilgiu, tačiau pagal atkūrimo metrikas yra aplenkti tik „VisionForce“ metodo.

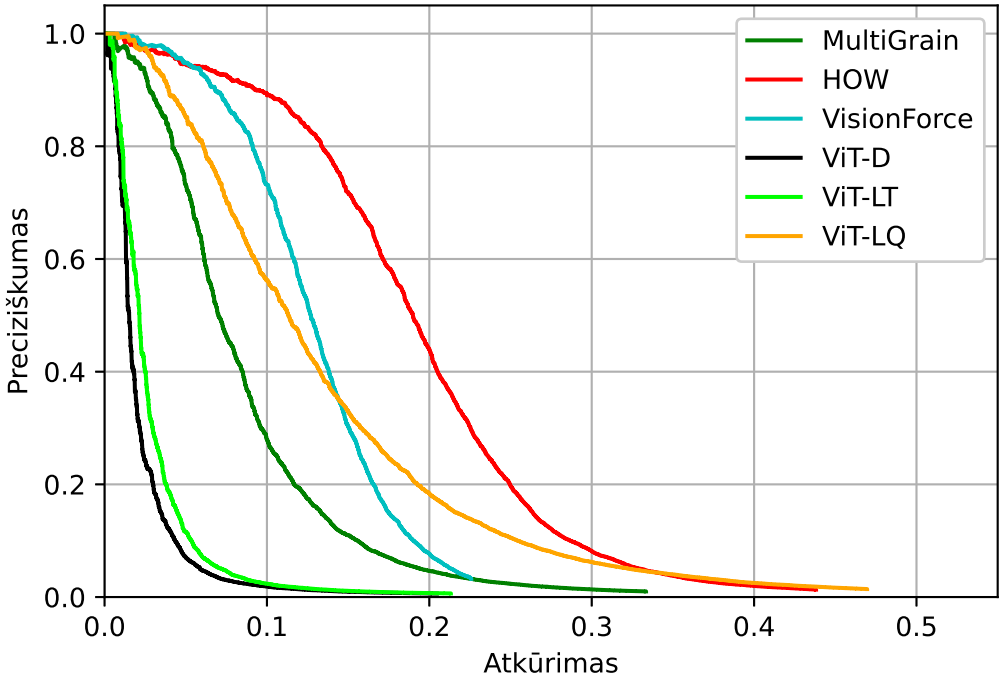


16 pav. Preciziškumo ir atkūrimo pokyčiai ViT-D, ViT-LT ir ViT-LQ modelių mokymo metu

16 paveikslėlyje matome preciziškumo ir atkūrimo įverčių pokyčius ViT-D, ViT-LT ir ViT-LQ modelių mokymo metu. Kaip matome pirmų dešimties epochų metu metrikų įverčiai laikosi panašiame lygyje. Nuo vienuoliktos epochos, pradedame naudoti itin sunkių atvejų paiešką. Matome visų metrikų pakilimą. Galime daryti išvadą, jog itin sunkių pavyzdžių paieška yra būtina aukštų įverčių pasiekimui. Taip pat matome tendenciją, jog mokymo metu, metrikų įverčiai pirma krenta, tuomet kyla. Detalūs metrikų įverčiai pateikiami priede nr. 2.

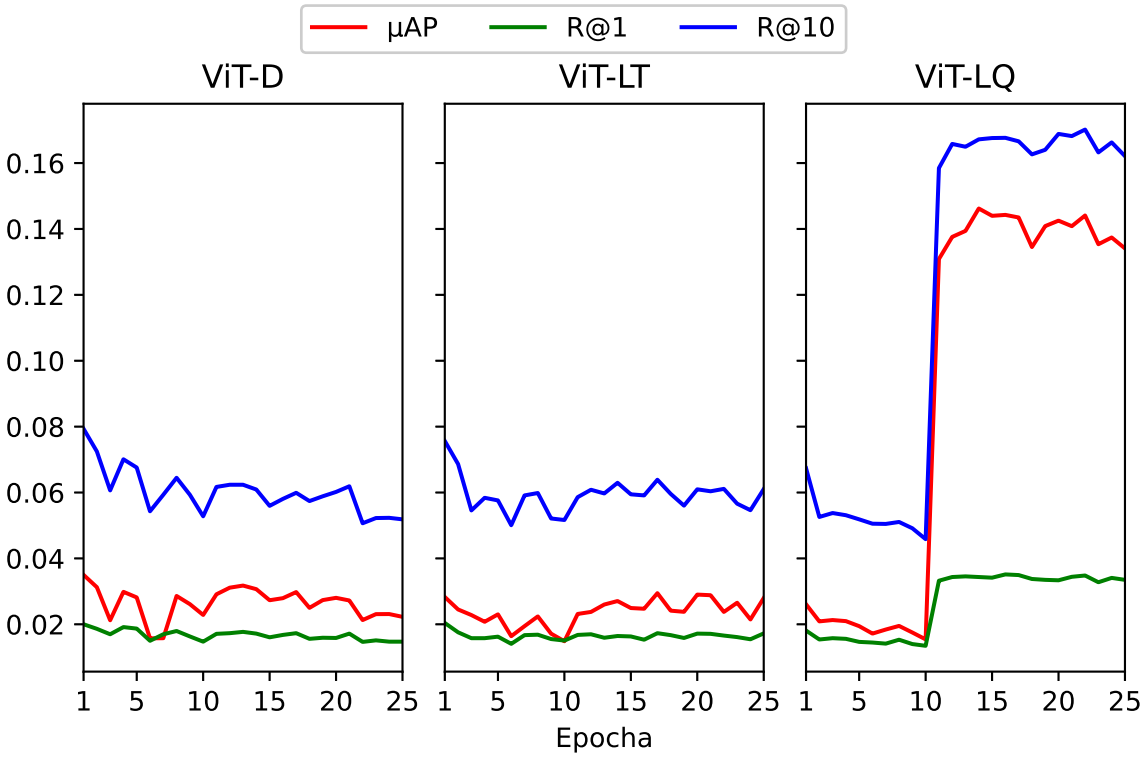
4 lentelė. Metodų vertinimo rezultatai, naudojant LAND duomenų rinkinį

Metodas	µAP	R@P=90	R@Rank1	R@Rank10
GIST	0,001	0,0001	0,003	0,009
MultiGrain	0,085	0,028	0,027	0,116
HOW	**0,192**	**0,095**	**0,043**	**0,206**
VisionForce	0,128	0,068	0,034	0,146
Klasifikacijos modelio pritaikymas (ResNetV2-101)	0,000	–	0,000	0,000
Klasifikacijos modelio pritaikymas (InceptionV3)	0,000	–	0,000	0,000
ViT-D	0,022	0,007	0,015	0,052
ViT-LT	0,028	0,007	0,017	0,061
ViT-LQ	0,134	0,039	0,033	0,162



17 pav. Metodų preciziškumo atkūrimo kreivės, naudojant LAND duomenų rinkinį

Iš duomenų, pateiktų 4 lentelėje ir 17 paveikslėlyje (pateikiami šeši geriausius rezultatus parodę metodai), matome kad geriausius rezultatus pasiekia HOW metodas. Preciziškumo pablogėjimas lemia jog „VisionForce“ metodas yra trečioje vietoje. Taip pat pastebime, kad apdorojant „Google Landmarks V2“ duomenų rinkinį suprastėjo visų jau egzistuojančių metodų rezultatai, nes šie metodai buvo pritaikyti spręsti DISC21 duomenų rinkinio problemą (kopijų paiešką), kuri skiriasi nuo LAND duomenų rinkinio problemos. Skirtingai nuo DISC21 duomenų rinkinio, atkūrimo, naudojant aukščiausią įvertį turintį atitikmens vaizdą, ir atkūrimo, naudojant 10 aukščiausią įvertį turinčių atitikmens vaizdų, metrikos ženkliai skiriasi. Naudojant 10 atitikmens vaizdų, atkūrimo įvertis pagerėja iki 5 kartų. Klasifikacijos modelio pritaikymo metodas išlieka paskutinėje vietoje. Darbo metu sukurtas vaizdų transformatoriais ir ketvertų tikslo funkcija paremtas modelis ViT-LQ yra antroje vietoje pagal preciziškumą ir atkūrimą, naudojant 10 vaizdų porų.



18 pav. Preciziškumo ir atkūrimo pokyčiai ViT-D, ViT-LT ir ViT-LQ modelių mokymo metu

18 paveikslėlyje matome preciziškumo ir atkūrimo įverčių pokyčius ViT-D, ViT-LT ir ViT-LQ modelių mokymo metu. Galima pastebėti, jog ViT-D ir ViT-LT mokymo metu metrikos kinta panašiai – vertinimo metrikos yra keičiamos, tačiau viso mokymo metu laikosi 3 % preciziškumo aplinkoje. Tai patvirtina hipotezę, jog tik vaizdų kopijų paieškai skirti metodai (mokyti naudojantis trejetų tikslo funkcija) netinka „Google Landmarks V2“ duomenų rinkinio užduoties sprendimui. Tuo tarpu ViT-LQ modelis, iš pradžių taip pat laikęsis 3 % preciziškumo aplinkoje, sparčiai pagerėja pradėjus naudoti itin sunkių atvejų paiešką. Detalūs metrikų įverčiai pateikiami priede nr. 3.

5 lentelė. Metodų našumo rezultatai

Metodas	Savybių išgavimo greitis (ms)	Porų paieškos greitis (mln. porų / s)
GIST	**6,7**	**21,977**
MultiGrain	20,4	13,557
HOW	118,5	0,120
VisionForce	313,5	1,611
Klasifikacijos modelio pritaikymas (ResNetV2-101)	31,3	17,702
Klasifikacijos modelio pritaikymas (InceptionV3)	26,7	18,308
ViT-D	23,1	19,608
ViT-LT	23,7	19,231
ViT-LQ	23,7	19,417

5 lentelėje pateikiami metodų našumo rezultatai (detalūs rezultatai pateikiami priede nr. 4). Greičiausiai veikia GIST metodas (nenaudojami neuroniniai tinklai), tačiau šio metodo rezultatai

nėra patenkinami. eriausius rezultatus parodę „HOW“ ir „VisionForce“ metodai pasižymi mažiausiu greičiu. Vaizdų transformatoriais paremti metodai yra antroje vietoje.

Rezultatai ir išvados

Darbo metu buvo tiriami 5 jau egzistuojantys vaizdų panašumo vertinimo metodai bei naujai pasiūlytas vaizdų transformatoriais paremtas metodas. Gauti šie rezultatai:

- Įvertinus vaizdų transformatorių variantus, pasirinkta naudoti ViT-L kaip mokomų modelių bazinę architektūrą.
- Geriausią preciziškumą, naudojant „DISC21“ duomenų rinkinį, parodė „VisionForce“ ir HOW metodai. Klasifikacijos modeliais paremtas metodas parodė prasčiausius rezultatus. Vaizdų transformatoriais paremti modeliai atsilieka nuo HOW, „VisionForce“ ir MultiGrain metodų μAP atžvilgiu, tačiau pagal atkūrimo metrikas yra antroje vietoje (tik „VisionForce“ metodo atkūrimo metrikos yra aukštesnės).
- Geriausią preciziškumą, naudojant „LAND“ duomenų rinkinį, parodė HOW metodas. Preciziškumo pablogėjimas lemia jog „VisionForce“ metodas yra trečioje vietoje. Skirtingai nuo DISC21 duomenų rinkinio, atkūrimo, naudojant aukščiausią įvertį turintį atitikmens vaizdą, ir atkūrimo, naudojant 10 aukščiausią įvertį turinčių atitikmens vaizdų, metrikos ženkliai skiriasi. Klasifikacijos modeliais paremtas metodas išlieka paskutinėje vietoje. Vaizdų transformatoriais ir ketvertų tikslo funkcija paremtas modelis ViT-LQ yra antroje vietoje pagal preciziškumą ir atkūrimą, naudojant 10 vaizdų porų.
- ViT-D, ViT-LT ir ViT-LQ modelių mokymo (naudojant „DISC21“ duomenų rinkinį) metu, pastebime, kad pirmų dešimties epochų metu metrikų įverčiai laikosi panašiame lygyje. Nuo vienuoliktos epochos, pradedame naudoti itin sunkių atvejų paiešką. Tai lemia visų metrikų pakilimą. Galime pastebėti, jog itin sunkių pavyzdžių paieška yra būtina aukštų įverčių pasiekimui.
- ViT-D, ViT-LT ir ViT-LQ modelių mokymo (naudojant „LAND“ duomenų rinkinį) metu, pastebime, jog ViT-D ir ViT-LQ mokymo metu metrikos kinta panašiai, t.y. yra keičiamos, tačiau viso mokymo metu laikosi 3 % preciziškumo aplinkoje. Tuo tarpu ViT-LQ modelis, iš pradžių taip pat laikęsis 3 % preciziškumo aplinkoje, sparčiai pagerėja pradėjus naudoti itin sunkių atvejų paiešką.
- Greičiausiai veikia GIST metodas. Geriausius rezultatus parodę „HOW“ ir „VisionForce“ metodai pasižymi mažiausiu greičiu. Vaizdų transformatoriais paremti metodai yra antroje vietoje.

Iš gautų rezultatų galime padaryti šias tyrimo išvadas:

- Geriausiais DISC21 duomenų rinkinio tyrime pasirodęs „VisionForce“ metodas yra pritaikytas būtent šio duomenų rinkinio užduočiai. Naudojant kitą duomenų rinkinį, metodo preciziškumas yra keletą kartų mažesnis. Dėl to modelis pasirodo blogiau už HOW metodą ir vaizdų transformatoriais paremtus metodus.

- Analizuojant „Google Landmarks V2“ duomenų rinkinį, visi jau egzistuojantys metodai pasirodė prasčiau. Darome išvadą, kad nei vienas šių modelių nėra pritaikytas bendram vaizdų panašumo vertinimo problemos sprendimui.
- Naivūs metodai, paremti klasifikacijos užduotims skirtais metodais, yra netinkami vaizdų panašumo vertinimo problemos sprendimui. Daroma išvada, jog šios problemos sprendimui būtinas specializuotas metodas.
- Tiek „DISC21“, tiek „LAND“ duomenų rinkinių naudojimo metu, vaizdų transformatoriais paremtų metodų naudojimas leidžia pasiekti aukštų rezultatų. Pasirinkta architektūra yra tinkama vaizdų panašumo vertinimo užduoties sprendimui.
- Sprendžiant tik vaizdų kopijų paieškos uždavinį („DISC21“ duomenų rinkinys), trejetų tikslo funkcija leidžia pasiekti aukštų rezultatų. Tačiau pakeitus duomenų rinkinį, toks mokymas yra nebetinkamas. Tuo tarpu naujai siūloma ketvertų tikslo funkcija, leidžia apmokyti vertinimo metodą spręsti tiek vaizdų kopijų paieškos, tiek panašių objektų paieškos uždavinius.
- Itin sunkių atvejų paieška yra būtinas procesas, norint pasiekti aukštų vertinimo rezultatų. Tai matome iš vertinimo metrikų pokyčių vaizdų transformatoriais paremtų metodų mokymo metu. Pradėjus naudoti sunkių atvejų paiešką, vertinimo metrikų įverčiai padidėja keletą kartų.

Norint pritaikyti siūlomą panašumo vertinimo metodą plataus masto naudojimui siūloma toliau mokyti metodą, naudojant ne dalinę, o pilną mokymo aibę, panagrinėti dinaminį mokymosi žingsnio keitimą.

Šaltiniai

[BJV+19] Maxim Berman, Hervé Jégou, Andrea Vedaldi, Iasonas Kokkinos ir Matthijs Douze. Multigrain: a unified image embedding for classes and instances. *arXiv preprint arXiv:1902.05509*, 2019.

[DBK+20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn ir k.t. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[DBP+20] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang ir Cristian Canton Ferrer. The deepfake detection challenge (dfdc) dataset. *arXiv preprint arXiv:2006.07397*, 2020.

[DCL+18] Jacob Devlin, Ming-Wei Chang, Kenton Lee ir Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[DTP+21] Matthijs Douze, Giorgos Tolias, Ed Pizzi, Zoë Papakipos ir k.t. The 2021 image similarity dataset and challenge. *arXiv preprint arXiv:2106.09672*, 2021.

[FS21] Anibal Fuentes ir Jose M Saavedra. Sketch-qnet: A quadruplet convnet for color sketch-based image retrieval. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 2134–2141, 2021.

[GAR+17] Albert Gordo, Jon Almazan, Jerome Revaud ir Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2):237–254, 2017.

[GR12] John Gantz ir David Reinsel. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. *IDC iView: IDC Analyze the future*, 2007(2012):1–16, 2012.

[GSA+20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec ir k.t. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[HLA+19] Sidra Hanif, Chao Li, Anis Alazzawe ir Longin Jan Latecki. Image Retrieval with Similar Object Detection and Local Similarity to Detected Objects. *Pacific Rim International Conference on Artificial Intelligence*, p. 42–55. Springer, 2019.

[HS20] Christof Henkel ir Philipp Singer. Supporting large-scale image recognition with out-of-domain samples. *arXiv preprint arXiv:2010.01650*, 2020.

[HZR+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren ir Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 770–778, 2016.

[JSB+20] Glenn Jocher, Alex Stoken, Jirka Borovec, Liu Changyu ir k.t. ultralytics/yolov5: v3. 1-bug fixes and performance improvements. *Version v3*, 1, 2020.

[KKH+22] Byungsoo Ko, Han-Gyu Kim, Byeongho Heo, Sangdoo Yun, Sanghyuk Chun, Geonmo Gu ir Wonjae Kim. Group Generalized Mean Pooling for Vision Transformer. *arXiv preprint arXiv:2212.04114*, 2022.

[KRA+20] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings ir k.t. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

[LLC+21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin ir Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF international conference on computer vision*, p. 10012–10022, 2021.

[LLS21] Zeming Li, Songtao Liu ir Jian Sun. Momentum^2 Teacher: Momentum Teacher with Momentum Statistics for Self-Supervised Learning. *arXiv preprint arXiv:2101.07525*, 2021.

[NAS+17] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand ir Bohyung Han. Large-scale image retrieval with attentive deep local features. *Proceedings of the IEEE international conference on computer vision*, p. 3456–3465, 2017.

[OT01] Aude Oliva ir Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

[OTG+99] Aude Oliva, Antonio B Torralba, Anne Guerin-Dugue ir Jeanny Herault. Global semantic classification of scenes using power spectrum templates, 1999.

[PB22] Zoe Papakipos ir Joanna Bitton. Augly: Data augmentations for robustness. *arXiv preprint arXiv:2201.06494*, 2022.

[PLS+18] Xingang Pan, Ping Luo, Jianping Shi ir Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. *Proceedings of the European Conference on Computer Vision (ECCV)*, p. 464–479, 2018.

[RPA14] Md Khalid Imam Rahmani, Naina Pal ir Kamiya Arora. Clustering of Image Data Using K-Means and Fuzzy K-Means. *International Journal of Advanced Computer Science and Applications*, 5(7), 2014.

[RTC18] Filip Radenović, Giorgos Tolias ir Ondřej Chum. Fine-tuning CNN image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1655–1668, 2018.

[SVI+16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens ir Zbigniew Wojna. Rethinking the inception architecture for computer vision. *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 2818–2826, 2016.

[SZ03] Josef Sivic ir Andrew Zisserman. Video Google: A text retrieval approach to object matching in videos. *Computer Vision, IEEE International Conference on*, tom. 3, p. 1470–1470. IEEE Computer Society, 2003.

[TAJ16] Giorgos Tolias, Yannis Avrithis ir Hervé Jégou. Image search with selective match kernels: aggregation across single and multiple images. *International Journal of Computer Vision*, 116(3):247–261, 2016.

[TJC20] Giorgos Tolias, Tomas Jenicek ir Ondřej Chum. Learning and aggregating deep local descriptors for instance-level recognition. *European Conference on Computer Vision*, p. 460–477. Springer, 2020.

[TO99] Antonio B Torralba ir Aude Oliva. Semantic organization of scenes using discriminant structural templates. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, tom. 2, p. 1253–1258. IEEE, 1999.

[TSF+16] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth ir Li-Jia Li. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

[UVG+13] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers ir Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser ir Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[WAC+20] Tobias Weyand, Andre Araujo, Bingyi Cao ir Jack Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, p. 2575–2584, 2020.

[WLZ+20] Wenhao Wang, Shengcai Liao, Fang Zhao, Cuicui Kang ir Ling Shao. Domainmix: Learning generalizable person re-identification without human annotations. *arXiv preprint arXiv:2011.11953*, 2020.

[WMS+17] Chao-Yuan Wu, R Manmatha, Alexander J Smola ir Philipp Krahenbuhl. Sampling matters in deep embedding learning. *Proceedings of the IEEE international conference on computer vision*, p. 2840–2848, 2017.

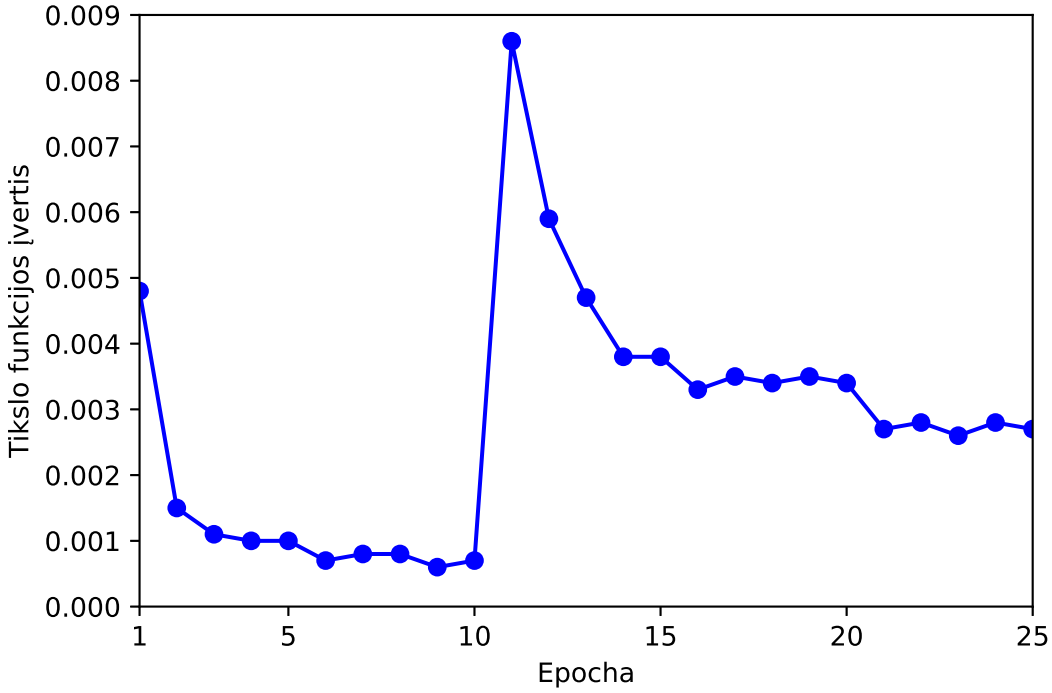
[WSZ+21] Wenhao Wang, Yifan Sun, Weipu Zhang ir Yi Yang. D^2LV: A Data-Driven and Local-Verification Approach for Image Copy Detection. *arXiv preprint arXiv:2111.07090*, 2021.

[WZL+22] Wenhao Wang, Fang Zhao, Shengcai Liao ir Ling Shao. Attentive waveblock: complementarity-enhanced mutual networks for unsupervised domain adaptation in person re-identification and beyond. *IEEE Transactions on Image Processing*, 31:1532–1544, 2022.

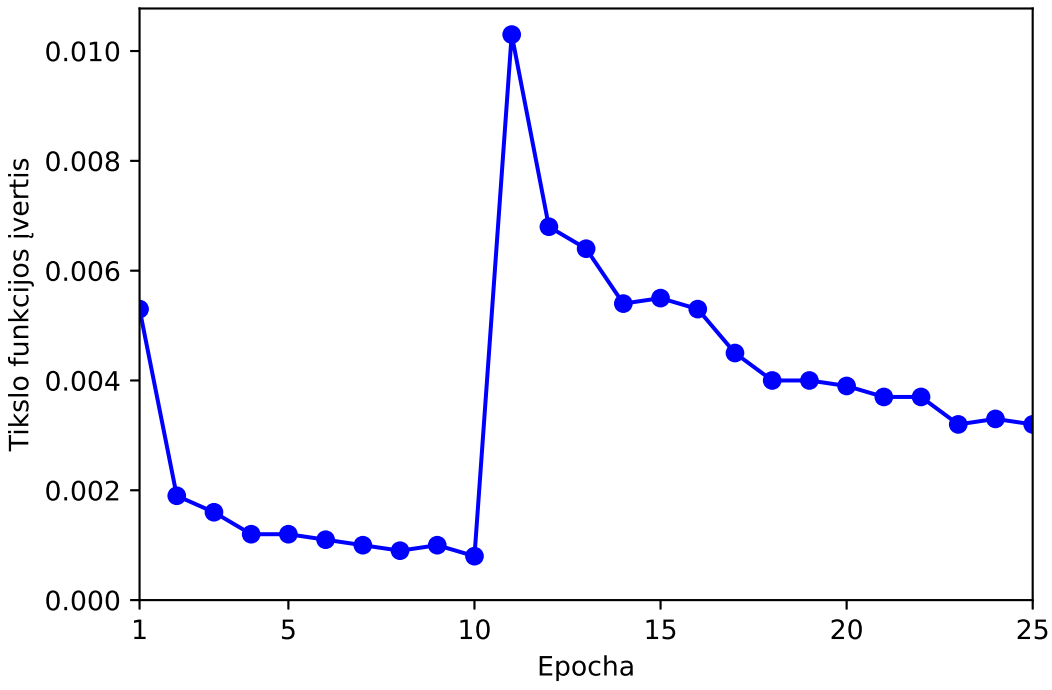
[ZJM+21] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun ir Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *International Conference on Machine Learning*, p. 12310–12320. PMLR, 2021.

Priedas nr. 1

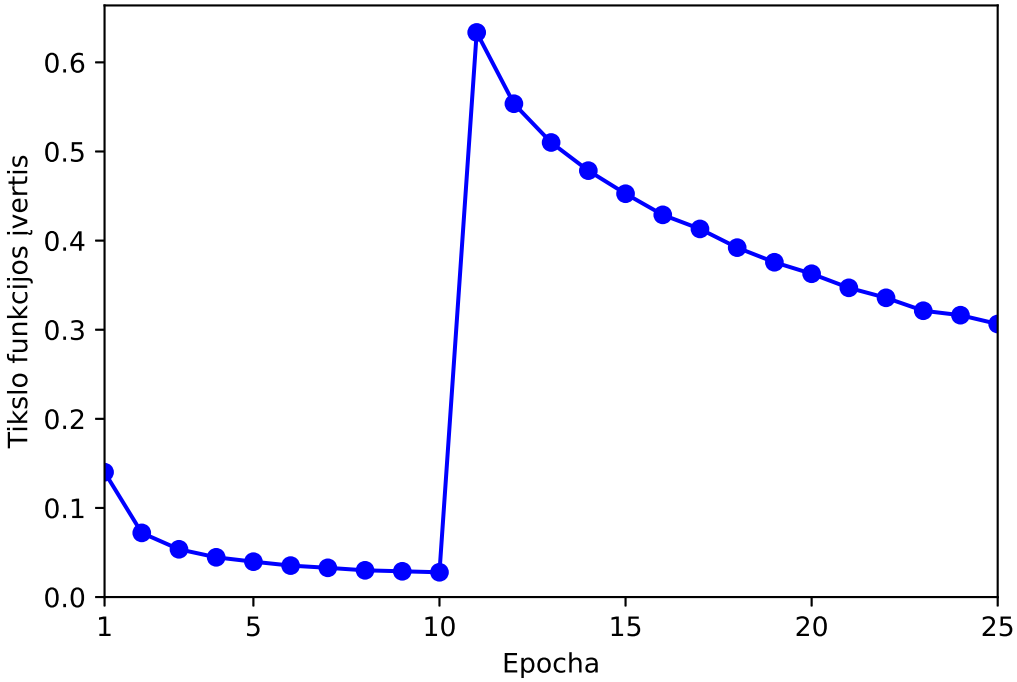
Tikslo funkcijų įverčiai skirtingose mokymo epochose



19 pav. ViT-D mokymo epochų tikslo funkcijų įverčiai



20 pav. ViT-LT mokymo epochų tikslo funkcijų įverčiai



21 pav. ViT-LQ mokymo epochų tikslo funkcijų įverčiai

Priedas nr. 2

Preziškumo ir atkūrimo pokyčiai ViT mokymo metu (DISC21)

6 lentelė. Preziškumo ir atkūrimo pokyčiai ViT-D mokymo metu

Epocha	µAP	R@P=90	R@Rank1	R@Rank10
01	0.17116	0.00251	0.43389	0.48567
02	0.17084	0.00201	0.44143	0.49372
03	0.15044	0.00201	0.43037	0.47109
04	0.18404	0.00101	0.46104	0.50830
05	0.14858	0.00050	0.46405	0.50679
06	0.08305	0.00201	0.42283	0.46154
07	0.11407	0.00503	0.42534	0.45249
08	0.18179	0.00452	0.45953	0.50277
09	0.18853	0.02815	0.44897	0.50427
10	0.15644	0.00050	0.44696	0.50377
11	0.24799	0.08145	0.50628	0.56058
12	0.23513	0.07240	0.50880	0.57014
13	0.23913	0.12469	0.51232	0.56310
14	0.24753	0.04274	0.53293	0.59175
15	0.26279	0.10256	0.51684	0.57567
16	0.23461	0.03369	0.50679	0.56662
17	0.22522	0.00201	0.49774	0.57416
18	0.18689	0.00151	0.50578	0.56008
19	0.20260	0.00302	0.48316	0.54651
20	0.24214	0.05581	0.49422	0.55003
21	0.28278	0.12117	0.50427	0.56511
22	0.25039	0.04827	0.51433	0.58019
23	0.31026	0.15586	0.50478	0.56410
24	0.30352	0.18049	0.50075	0.56360
25	0.28337	0.16893	0.49070	0.55455

7 lentelė. Preziškumo ir atkūrimo pokyčiai ViT-LT mokymo metu

Epocha	µAP	R@P=90	R@Rank1	R@Rank10
01	0.21432	0.05229	0.45048	0.48768
02	0.09827	0.00151	0.43590	0.47059
03	0.17786	0.00201	0.46204	0.51383
04	0.13057	0.00201	0.44394	0.47813
05	0.13306	0.00151	0.44143	0.48819
06	0.13653	0.00101	0.44093	0.48718
07	0.12680	0.00101	0.44997	0.48919
08	0.10733	0.00151	0.44545	0.48617
09	0.10926	0.00251	0.41730	0.45400
10	0.06695	0.00302	0.39970	0.44646
11	0.16103	0.00201	0.49070	0.52388
12	0.15313	0.00151	0.48165	0.52740
13	0.17748	0.00251	0.47763	0.53494
14	0.20002	0.00251	0.51031	0.56058
15	0.15770	0.00201	0.47662	0.52388
16	0.14439	0.00101	0.48014	0.52740
17	0.22768	0.05882	0.50980	0.55958
18	0.17598	0.00402	0.48969	0.53394
19	0.17149	0.00302	0.49522	0.54952
20	0.23424	0.04525	0.50277	0.55958
21	0.21296	0.06938	0.48969	0.54198
22	0.10900	0.00151	0.47562	0.51986
23	0.16937	0.00452	0.49271	0.54751
24	0.15799	0.00201	0.48215	0.53544
25	0.17246	0.01810	0.48869	0.54651

8 lentelė. Preziškumo ir atkūrimo pokyčiai ViT-LQ mokymo metu

Epocha	µAP	R@P=90	R@Rank1	R@Rank10
01	0.24217	0.16189	0.44796	0.47813
02	0.22555	0.12217	0.45651	0.49522
03	0.21671	0.12469	0.43992	0.47813
04	0.22263	0.13072	0.44042	0.48617
05	0.19426	0.11815	0.40774	0.45199
06	0.16501	0.10709	0.41931	0.46405
07	0.21698	0.13122	0.45249	0.49522
08	0.22867	0.14279	0.43841	0.49975
09	0.14728	0.05028	0.37054	0.41579
10	0.18543	0.11865	0.39970	0.44746
11	0.15029	0.08798	0.41579	0.45701
12	0.18942	0.11865	0.43439	0.48819
13	0.20966	0.13776	0.44796	0.50327
14	0.16510	0.07491	0.42936	0.48366
15	0.18857	0.12267	0.43288	0.48617
16	0.22037	0.14580	0.45601	0.51433
17	0.17878	0.10508	0.41629	0.48316
18	0.18029	0.09050	0.43238	0.49271
19	0.17767	0.10106	0.45249	0.50830
20	0.17093	0.09955	0.42584	0.48416
21	0.18783	0.10508	0.44193	0.50075
22	0.19393	0.11111	0.44143	0.50729
23	0.21522	0.13575	0.44595	0.50980
24	0.21431	0.11966	0.47210	0.52690
25	0.18914	0.09351	0.45902	0.51433

Priedas nr. 3

Preziškumo ir atkūrimo pokyčiai ViT mokymo metu (LAND)

9 lentelė. Preziškumo ir atkūrimo pokyčiai ViT-D mokymo metu

Epocha	µAP	R@P=90	R@Rank1	R@Rank10
01	0.03500	0.00773	0.02002	0.07934
02	0.03126	0.00674	0.01863	0.07247
03	0.02122	0.00205	0.01698	0.06064
04	0.02986	0.00667	0.01916	0.07009
05	0.02816	0.00661	0.01869	0.06758
06	0.01574	0.00277	0.01500	0.05430
07	0.01578	0.00178	0.01698	0.05932
08	0.02860	0.00654	0.01797	0.06447
09	0.02612	0.00634	0.01632	0.05932
10	0.02284	0.00680	0.01473	0.05278
11	0.02911	0.00733	0.01711	0.06170
12	0.03112	0.00793	0.01731	0.06236
13	0.03176	0.00859	0.01770	0.06236
14	0.03066	0.00938	0.01718	0.06091
15	0.02731	0.00740	0.01605	0.05595
16	0.02793	0.00773	0.01678	0.05807
17	0.02978	0.00839	0.01731	0.05992
18	0.02500	0.00746	0.01559	0.05741
19	0.02735	0.00746	0.01592	0.05886
20	0.02804	0.00872	0.01585	0.06018
21	0.02723	0.00647	0.01718	0.06190
22	0.02129	0.00542	0.01467	0.05067
23	0.02309	0.00713	0.01513	0.05225
24	0.02311	0.00680	0.01473	0.05232
25	0.02227	0.00674	0.01473	0.05186

10 lentelė. Preziškumo ir atkūrimo pokyčiai ViT-LT mokymo metu

Epocha	µAP	R@P=90	R@Rank1	R@Rank10
01	0.02829	0.00832	0.02041	0.07564
02	0.02451	0.00423	0.01757	0.06864
03	0.02281	0.00740	0.01579	0.05456
04	0.02077	0.00324	0.01579	0.05840
05	0.02303	0.00542	0.01625	0.05760
06	0.01640	0.00383	0.01407	0.05007
07	0.01944	0.00258	0.01671	0.05912
08	0.02240	0.00244	0.01685	0.05985
09	0.01716	0.00172	0.01552	0.05212
10	0.01486	0.00185	0.01506	0.05166
11	0.02315	0.00555	0.01678	0.05859
12	0.02376	0.00423	0.01698	0.06084
13	0.02600	0.00674	0.01592	0.05972
14	0.02706	0.00621	0.01645	0.06295
15	0.02494	0.00773	0.01632	0.05945
16	0.02472	0.00357	0.01533	0.05912
17	0.02945	0.00898	0.01731	0.06388
18	0.02414	0.00601	0.01671	0.05959
19	0.02379	0.00449	0.01585	0.05602
20	0.02902	0.00839	0.01718	0.06097
21	0.02883	0.00641	0.01711	0.06038
22	0.02375	0.00337	0.01658	0.06110
23	0.02655	0.00641	0.01612	0.05661
24	0.02147	0.00436	0.01546	0.05463
25	0.02798	0.00727	0.01724	0.06104

11 lentelė. Preziškumo ir atkūrimo pokyčiai ViT-LQ mokymo metu

Epocha	μAP	R@P=90	R@Rank1	R@Rank10
01	0.02609	0.00832	0.01810	0.06738
02	0.02089	0.00469	0.01539	0.05258
03	0.02129	0.00548	0.01579	0.05377
04	0.02097	0.00628	0.01559	0.05311
05	0.01944	0.00476	0.01467	0.05186
06	0.01718	0.00502	0.01447	0.05054
07	0.01841	0.00495	0.01414	0.05047
08	0.01951	0.00423	0.01533	0.05106
09	0.01751	0.00436	0.01400	0.04915
10	0.01543	0.00416	0.01348	0.04584
11	0.13087	0.04380	0.03323	0.15848
12	0.13759	0.04736	0.03435	0.16581
13	0.13941	0.04822	0.03455	0.16495
14	0.14621	0.04525	0.03435	0.16720
15	0.14398	0.04783	0.03415	0.16759
16	0.14427	0.05139	0.03514	0.16766
17	0.14348	0.04697	0.03495	0.16660
18	0.13451	0.04168	0.03376	0.16264
19	0.14087	0.04822	0.03349	0.16402
20	0.14252	0.04347	0.03336	0.16885
21	0.14084	0.04512	0.03442	0.16819
22	0.14408	0.04875	0.03481	0.17017
23	0.13536	0.04162	0.03277	0.16323
24	0.13741	0.04049	0.03409	0.16627
25	0.13410	0.03944	0.03349	0.16211

Priedas nr. 4
Detalios metodų našumo metrikos

Čia trukmė pateikiama sekundėmis, o greitis – sekundėmis per vaizdą (porų paieškos greitis – mln. vaizdų per sekundę).

12 lentelė. GIST metodo našumas, naudojant DISC21 duomenų rinkinį

Procesas	Bendra trukmė	Vaizdų kiekis	Greitis
PCA mokymas	30,5071	4000	0,0076
Užklausos vaizdų savybių gavimas	67,0061	10000	0,0067
Atitikmens vaizdų savybių gavimas	1361,3014	100000	0,0136
Atitikmens vaizdų savybių gavimas (naudojant 5000 vaizdų rinkinius)	1361,3014	100000	0,0074
Porų paieška	45,503	10000 × 100000	21,977

13 lentelė. GIST metodo našumas, naudojant LAND duomenų rinkinį

Procesas	Bendra trukmė	Vaizdų kiekis	Greitis
Užklausos vaizdų savybių gavimas	6,1215	1129	0,0054
Atitikmens vaizdų savybių gavimas	54,4859	10000	0,0054
Porų paieška	0,6434	1129 × 10000	17,546

14 lentelė. „MultiGrain“ metodo našumas, naudojant DISC21 duomenų rinkinį

Procesas	Bendra trukmė	Vaizdų kiekis	Greitis
PCA mokymas	199,0715	10000	0,0199
Užklausos vaizdų savybių gavimas	168,9528	10000	0,0169
Atitikmens vaizdų savybių gavimas	1737,4650	100000	0,0174
Mokymo vaizdų savybių gavimas	1736,1167	100000	0,0174
Normalizacija	3,3370	10000	0,0033
Porų paieška	73,762	10000 × 100000	13,557

15 lentelė. „MultiGrain“ metodo našumas, naudojant LAND duomenų rinkinį

Procesas	Bendra trukmė	Vaizdų kiekis	Greitis
Užklausos vaizdų savybių gavimas	21,6197	1129	0,0191
Atitikmens vaizdų savybių gavimas	177,4440	10000	0,0177
Normalizacija	1,4129	1129	0,0013
Porų paieška	0,87275	1129 × 10000	12,936

16 lentelė. HOW metodo našumas, naudojant DISC21 duomenų rinkinį

Procesas	Bendra trukmė	Vaizdų kiekis	Greitis
Atitikmens vaizdų savybių gavimas (naudojant 10000 vaizdų rinkinius)	11175,2341	100000	0,1118
Atitikmens vaizdų IVF failo kūrimas	666,5788	100000	0,0067
Porų paieška (atitikmens vaizdų aibė, naudojant 500 vaizdų rinkinius)	8302,0961	10000 × 100000	0,120
Mokymo vaizdų savybių gavimas (naudojant 10000 vaizdų rinkinius)	11362,1299	100000	0,1136
Mokymo vaizdų IVF failo kūrimas	843,7129	100000	0,0084
Porų paieška (mokymo vaizdų aibė, naudojant 500 vaizdų rinkinius)	7313	10000 × 100000	0,137

17 lentelė. HOW metodo našumas, naudojant LAND duomenų rinkinį

Procesas	Bendra trukmė	Vaizdų kiekis	Greitis
Atitikmens vaizdų savybių gavimas	886,6985	10000	0,0887
Atitikmens vaizdų IVF failo kūrimas	89,5448	10000	0,0090
Porų paieška atitikmens vaizdų aibėje	401,0296	1129 × 10000	0,028
Porų paieška mokymo vaizdų aibėje	751,437238	1129 × 10000	0,015

18 lentelė. „VisionForce“ augmentacijų metrikos, naudojant DISC21 duomenų rinkinį

Procesas	Bendra trukmė	Vaizdų kiekis	Greitis
Užklausos vaizdų aibės augmentacija	11846,45838	10000	1,1185
Atitikmens vaizdų aibės augmentacija	14591,885747	100000	0,1460

19 lentelė. „VisionForce“ augmentacijų metrikos, naudojant LAND duomenų rinkinį

Procesas	Bendra trukmė	Vaizdų kiekis	Greitis
Užklausos vaizdų aibės augmentacija	4092,9615	1129	3,6253
Atitikmens vaizdų aibės augmentacija	1216,8017	10000	0,1217

20 lentelė. „VisionForce“ metodo našumas, naudojant DISC21 duomenų rinkinį

Procesas	Bendra trukmė	Vaizdų kiekis	Greitis
Augmentuotų užklausų vaizdų savybių gavimas (vaizdų dydis – 256)	33664,9157658	319848	0,1053
Originalių atitikmens vaizdų savybių gavimas	17647,0611	100000	0,1765
PCA mokymas (vaizdų dydis – 256)	20877,9316	20000	1,0439
Mokymo vaizdų savybių gavimas (vaizdų dydis – 256)	21720,5440	100000	0,2172
Augmentuotų užklausų vaizdų savybių gavimas (vaizdų dydis – 200)	20528,31748	319848	0,0642
Originalių atitikmens vaizdų savybių gavimas (vaizdų dydis – 200)	9652,5017	100000	0,0965
PCA mokymas (vaizdų dydis – 200)	19810,2066	20000	0,9905
Mokymo vaizdų savybių gavimas (vaizdų dydis – 200)	9293,2680	100000	0,0929
Augmentuotų užklausų vaizdų savybių gavimas (vaizdų dydis – 320)	46070,3244	319848	0,1440
Originalių atitikmens vaizdų savybių gavimas (vaizdų dydis – 320)	25942,53573	100000	0,2594
PCA mokymas (vaizdų dydis – 320)	24016,2283	20000	1,2008
Mokymo vaizdų savybių gavimas (vaizdų dydis – 320)	26099,7637	100000	0,2601
Porų paieška	620,555	10000×100000	1,611

21 lentelė. „VisionForce“ metodo našumas, naudojant LAND duomenų rinkinį

Augmentuotų užklausų vaizdų savybių gavimas (vaizdų dydis – 256)	2149,160606	29938	0,0718
Originalių atitikmens vaizdų savybių gavimas (vaizdų dydis – 256)	1381,927342	10000	0,1382
Augmentuotų užklausų vaizdų savybių gavimas (vaizdų dydis – 200)	2503,886304	29938	0,0836
Originalių atitikmens vaizdų savybių gavimas (vaizdų dydis – 200)	1967,272363	10000	0,1967
Augmentuotų užklausų vaizdų savybių gavimas (vaizdų dydis – 320)	4014,4748	29938	0,1341
Originalių atitikmens vaizdų savybių gavimas (vaizdų dydis – 320)	2487,3530	10000	0,2487
Porų paieška	38,649	1129×10000	0,292

22 lentelė. Klasifikacijai pritaikyto metodo našumas, naudojant DISC21 duomenų rinkinį

Procesas	Bendra trukmė	Vaizdų kiekis	Greitis
Užklausų vaizdų savybių gavimas (ResnetV2)	312,808881	10000	0,0313
Užklausų vaizdų savybių gavimas (InceptionV3)	266,783155	10000	0,0267
Atitikmens vaizdų savybių gavimas (ResnetV2)	3767,911434	100000	0,0377
Atitikmens vaizdų savybių gavimas (InceptionV3)	3158,4563	100000	0,0316
Mokymo vaizdų savybių gavimas (ResnetV2)	3777,9726	100000	0,0378
Mokymo vaizdų savybių gavimas (InceptionV3)	3206,9665	100000	0,0321
Porų paieška (ResnetV2)	56,489213	10000 $\times$ 100000	17,702
Porų paieška (InceptionV3)	54,621582	10000 $\times$ 100000	18,308

23 lentelė. Klasifikacijai pritaikyto metodo našumas, naudojant LAND duomenų rinkinį

Procesas	Bendra trukmė	Vaizdų kiekis	Greitis
Užklausų vaizdų savybių gavimas (ResnetV2)	112,3358	1129	0,0995
Užklausų vaizdų savybių gavimas (InceptionV3)	262,611355	1129	0,0267
Atitikmens vaizdų savybių gavimas (ResnetV2)	1034,010785	10000	0,1034
Atitikmens vaizdų savybių gavimas (InceptionV3)	2105,432387	100000	0,0316
Porų paieška (ResnetV2)	0,966522	1129$\times$10000	11,681
Porų paieška (InceptionV3)	0,800729	1129$\times$10000	14,100

24 lentelė. Vaizdų transformatoriais paremto metodo našumas, naudojant DISC21 duomenų rinkinį

Procesas	Bendra trukmė	Vaizdų kiekis	Greitis
Užklausų vaizdų savybių gavimas (ViT-D)	437	10000	0,0437
Užklausų vaizdų savybių gavimas (ViT-LT)	650	10000	0,0650
Užklausų vaizdų savybių gavimas (ViT-LQ)	594	10000	0,0594
Atitikmens vaizdų savybių gavimas (ViT-D)	2957	100000	0,0296
Atitikmens vaizdų savybių gavimas (ViT-LT)	6152	100000	0,0615
Atitikmens vaizdų savybių gavimas (ViT-LQ)	4638	100000	0.0464
Mokymo vaizdų savybių gavimas (ViT-D)	5922	100000	0,0592
Porų paieška (ViT-D)	51	10000 $\times$ 100000	19,231
Porų paieška (ViT-LT)	52	10000 $\times$ 100000	19,417
Porų paieška (ViT-LQ)	51,5	10000 $\times$ 100000	14,609

25 lentelė. Vaizdų transformatoriais paremto metodo našumas, naudojant LAND duomenų rinkinį

Procesas	Bendra trukmė	Vaizdų kiekis	Greitis
Užklausų vaizdų savybių gavimas (ViT-D)	37	1129	0,0328
Užklausų vaizdų savybių gavimas (ViT-LT)	66	1129	0,0650
Užklausų vaizdų savybių gavimas (ViT-LQ)	33	1129	0,0300
Atitikmens vaizdų savybių gavimas (ViT-D)	402	10000	0,0402
Atitikmens vaizdų savybių gavimas (ViT-LT)	551	10000	0,0551
Atitikmens vaizdų savybių gavimas (ViT-LQ)	363	10000	0.0363
Mokymo vaizdų savybių gavimas (ViT-LT)	6327	138520	0,0457
Mokymo vaizdų savybių gavimas (ViT-LQ)	6750	138520	0,0487
Porų paieška (ViT-D)	0,901	1129×10000	12,530
Porų paieška (ViT-LT)	0,867	1129×10000	13,022
Porų paieška (ViT-LQ)	0,814	1129×10000	13,874