

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Detalus vaizdų panašumas naudojant trejetų tinklus

Fine-grained images similarity using triplet-network

Kursinis darbas

Atliko:	4 kurso 6 grupės studentas	
	Andrius Butkevičius	(parašas)
Darbo vadovas:	dr. Vytautas Valaitis	(parašas)

Vilnius – 2020

TURINYS

ĮVADAS	2
TIKSLAS	3
UŽDAVINIAI	3
1. TREJETŲ IR SIAMESE TINKLŲ TYRIMŲ LITERATŪROS ANALIZĖ	4
1.1. Gilieji konvoliuciniai neuronų tinklai	4
1.2. Trejetų tinklai.....	4
1.2.1. Architektūra	5
1.3. Siamese tinklas	6
1.3.1. Architektūra	6
1.4. Nefiksuoto dydžio vaizdai konvoliuciniuose tinkluose	7
2. ĮVERTINIMO FUNKCIJA	8
2.1. Treniruojami duomenys.....	9
3. TREJETŲ TINKLO VAIZDŲ ATPAŽINIMO TYRIMAS IR VERTINIMAS	10
3.1. Kaip išspręsti semantines vazidų skirtumo problemas	10
3.2. Treniravimo ir testavimo aplinka	10
3.3. VGG16 tinklo modelis	10
3.4. Tyrimo rezultatai	10
3.5. Vaizdai su prastais rezultatų palyginimais	11
3.6. Sprendimo būdai	13
4. REZULTATAI IR IŠVADOS	14
4.1. Rezultatai	14
4.2. Išvados.....	14
4.3. Rekomendacijos ateities darbams	14
5. PRIEDAI	15
5.1. Žodynas.....	15
LITERATŪRA	16

Įvadas

Vaizdų panašumo įvertinimas ir lyginimas tampa vis plačiau naudojamas ir susilaukia vis didesnio dėmesio informacinių technologijų srityje. Visgi plėtojant šią technologiją ir siekiant išgauti kuo korektiškesnius rezultatus yra susiduriama su problemomis, pavyzdžiui.: kaip kompiuteriui efektyviau atskirti vizualius skirtumus ir vaizdų panašumus žinant, kad kompiuterinėms mašinoms, tai nėra taip lengva, kaip palyginus žmogui, kuris gali akimirksniu sugebėti atpažinti aplink jį esamus objektus. Vaizdų identifikavimas ir jų lyginimas plačiai naudojama vaizdų klasifikavime, objektų atpažinime, pvz.: vaizdinės tapatybės nustatymui, veido atpažinimui, esamos vietovės aptikimui pasitelkiant beipiločio orlaivio užfiksuotas reljefo nuotraukų, nuotraukų radimui pasitelkiant paieškos sistemas. Norint sugebėti tai atpažinti ir klasifikuoti, yra pasitelkiami įvairūs tinklų modeliai ir jie treniruojami, tokie kaip Siamese arba trejetų tinklai. Šie tinklai susideda iš dviejų ar trijų vienetų ir lygiagrečių konvoliucinių neurono tinkle esančių atšakų, kurios tarpusavyje dalinasi sveriais, kurių dėka galima gauti aukšto lygio nuotraukų bruožų atvaizdavimą taip leidžiant panašioms nuotraukoms būti kuo arčiau sujungtoms viena su kita funkcijos erdvėje. Tuo metu nepanašioms vaizdams – būnant kuo toliau nuo teisingų nuotraukų. Rinkoje yra siūlomi įvairūs konvoliucinių neuronų tinklų modeliai, pvz.: Alexnet, VGGNet, GoogLeNet, ResNet. Verta paminėti, kad visi šie modeliai turėjo efektyvius atpažinimo rezultatus ILSVRC. Taigi vis dažniau yra pasitelkiama prieš tai minėti trejetų tinklai, kurie optimizuoja bruožų atstumus erdvėje. Visgi sudėtingiausia problema išlieka dėl semantinio tarpo problemos, kuri atsiranda tarp žemos rezoliucijos nuotraukos pikselių užfiksuotų kompiuterinių sistemų ir aukšto lygio semantinių konceptų, kurias suvokia žmonės. Dėl to reikia rasti geresnių būdų, kaip sugebėti pateikti nuotrauką kompiuteriui ir gauti gilesnias jos semantinius bruožus. Todėl šio darbo tikslas yra ištirti esamus rinkoje modelių veikimo principus, jų niuansus bei palyginimą vieną su kitu. Išanalizuoti, kada, kokiomis sąlygomis jie veikia geriausiai. Taigi bus pasirinkta viena iš esamų vaizdų duomenų rinkinių ir bus treniruojamas trejetų tinklas, analizuojamos rezultatai, problemos.

Tikslas

Šio darbo tikslas yra:

- Gautus vaizdų palyginimo rezultatus paanalizuoti detaliau, išskirti jų metrikas.
- Palyginti architektūrą su kitais neuronų tinklų modeliais.

Uždaviniai

1. Ištreniruoti trejetų tinklų modelį atpažįstant vaizdus.
2. Išskirti metrikas pagal kurias galėtų analizuoti gautus rezultatus, panaudojant trejetų tinklų modelį.
3. Išskirti esamų tyrimų trūkumus ir galimas sritis ateities darbams.
4. Palyginti kitų neuronų tinklų modelių architektūras.

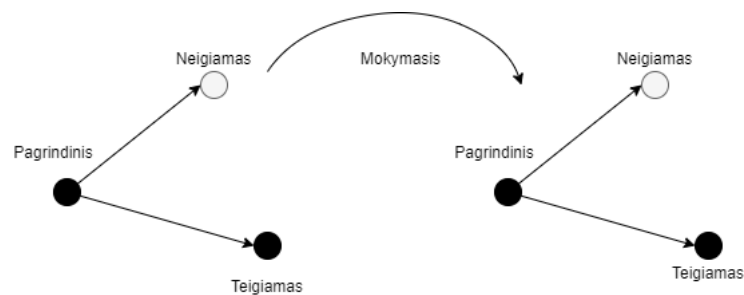
1. Trejetų ir Siamese tinklų tyrimų literatūros analizė

1.1. Gilieji konvoliuciniai neuronų tinklai

Giliajame mokymese, šis tinklas yra klasė iš giliųjų neuronų tinklų, kuri dažniausiai naudojama pritaikant vaizdų atpažinimą. Tai yra algoritmas, kuris įvestyje pasiima nuotrauką, priskiria jai svorius ir tendencijas jos bruožų įvairiose nuotraukos dalyse ir sugeba pagal tai atskirti panašumus lyginant su kitomis nuotraukomis.

1.2. Trejetų tinklai

Trejetų tinklų modelis buvo pasiulytas 2014m., moksliniko Chang Wang[JYT⁺14]. Modelio veikimo principas susideda iš trijų identiškų konvoliucinių neuroninių tinklų šakų, kurie tapusavyje dalijasi gautais svoriais. Kai trejetų tinklas įvestyje gauna tris pavyzdžius, išvestyje yra grąžinama dvejų tarpinės reikšmės, kurios nurodo atstumus lyginimus su pagrindine įvestimi ir kitomis dvejomis įvestimis, kurių rezultatas yra vektoriai.



1 pav. Trejeto tinklo mokymosi procesas

1.2.1. Architektūra

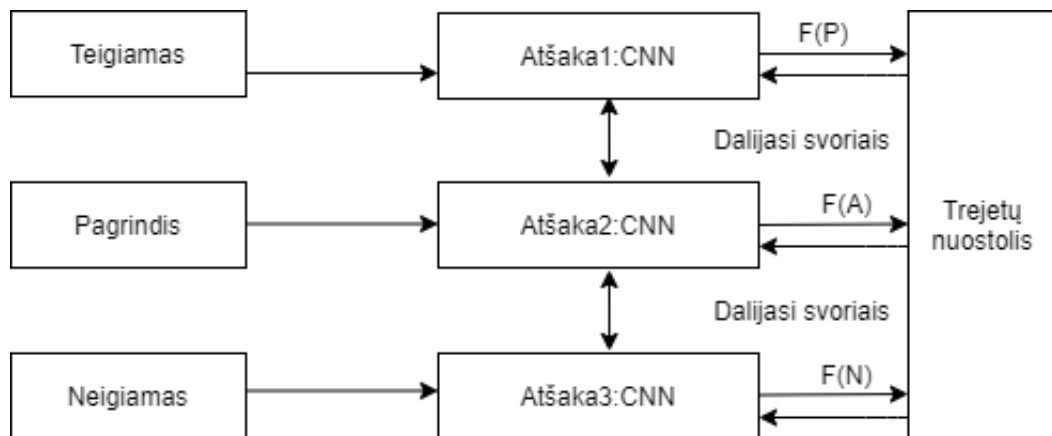
Trejetų tinklų modelis įvedimo dalyje reikalauja trijų įvesčių tuo pačiu metu. Kiekvienas jų turi savo pavadinimą ir reikšmę tinkle, pagrindinis x^a , pozityvus x^p , neigiamas x^n . Įvesties poros x^a ir x^p yra tos pačios kategorijos arba panašūs semantiškai įvestys. Tuo metu x^a ir x^n yra skirtingos kategorijos arba nepanašūs visai. Pasitelkiant funkcijas yra apskaičiuojama jų semantiniai panašumai atsumo erdvėje (nes jie yra pateikiami kaip vektoriai). Ši funkcija yra vadinama nuostolių funkcija, jos funkcija aprašyta žemiau. Kur parametras α rodo tarpą tarp x^a ir x^p bei x^a ir x^n . N reiškia skaičių nusakantį kiek trejetų įvesčių yra pateikiama. Tikslas yra pasiekti, kad x^a ir x^p būtų mažiau nei x^a ir x^n .

$$loss = \sum_{x=1}^N \max(d(x^a, x^p) - d(x^a, x^n) + \alpha, 0)$$

Kur d yra atstumo metrika, taip kad $d(x^a, x^p) < d(x^a, x^n)$, o N yra skaičius trejetų.

Alternatyvios nuostolių funkcijos. Artima šiai funkcijai yra ši lygtis. Ji naudoja kvadratinį Euklido atstumą kaip atstumo metriką ir $\alpha > 0$, nurodanti ribą tarp dviejų vaizdų ir absoliučios sumos reikšmės.

$$loss = \sum_{x=1}^N [||x^a - x^p|| - ||x^a - x^n|| + \alpha]$$



2 pav. Trejeto tinklų veikimo principas kartu su neuroninių tinklų lygiagrečiomis atšakomis

Šis neuroninių tinklų architektūrinis sprendimas padeda apmokyti duomenų klasifikavimą išskaidant duomenis pagal panašumus ir skirtumus. Jų metu keli lygiagretūs gileji neuroniniai tinklai yra apmokami bei tuo pačiu metu jie dalijasi svoriais vieni su kitu treniruojant tinklą. Trejetų tinkle tikslas yra sukurti trejetus, kurie susideda iš pagrindinio p^a , teigiamio x^p , neigiamio p^n – įvesties elementų. Pagrindinis elementas, tai kažkokia įvestis (nuotrauka, paveikslukas, muzikos įrašas ir t.t), kuriai mes bandome rasti atitikimą iš kitos įvesties. Teigiamą įvestis – panašus elementas atitinkamai pagrindiniai nuotraukai. Tuo tarpu neigiamas elementas skaitosi tas, kuris neturi panašumų su pagrindine nuotrauka. Neuroniniai tinklai apskaičiuoja

$\pi : f(\pi) \in$ Šie trys elementai yra įvedami nepriklausomai vienas nuo kito į tris identiškus giluosius neuroninius tinklus, kurie dalijasi vienoda architektūra ir parametrais. Jų metu yra apskaičiuojami atstumai tarp šių elementų. Šiam apskaičiavimui yra naudojama praradimo funkcija. Įmanomi rezultatai trjetų tinkle:

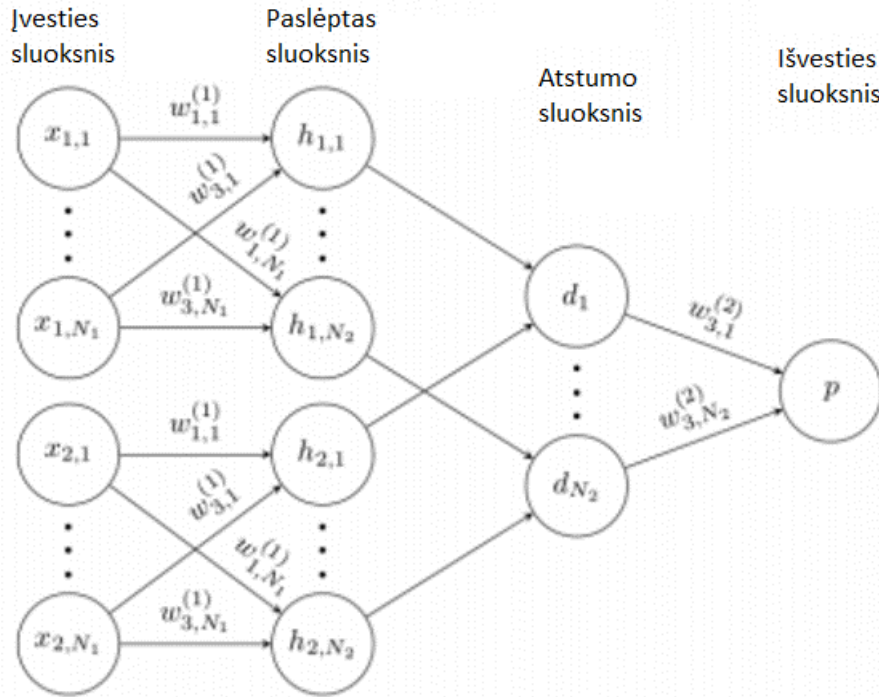
- Lengvas trejetas: trejetai, kurie turi nuostolį, reikšmę 0, nes $d(x^a, x^p) + \alpha < d(x^a, x^p)$
- Sudėtingas trejetas: trejetai, kurių neigiamumas yra arčiau pagrindinio negu teigiamų, $d(x^a, x^p) < d(x^a, x^p)$
- Pusiau sudėtingas trejetas: trejetai, kurių neigiamumas nėra arčiau teigiamumo, tačiau vis tiek turi teigiamą nuostolį: $d(x^a, x^n) < d(x^a, x^p) + \alpha$

1.3. Siamese tinklas

Pirma kartą publikuotas 1990 metais, autorių Bromely ir Yann LeCun, siekiant išspręsti parašo verifikavimo problema kaip vaizdo atitikimo problemą[BGL⁺90].

1.3.1. Architektūra

Tinklo modelis – neuronų tinklas turintis kelis ar daugiau identiškų neuroninių tinkle atšakas su vienodais parametrais[MKR16]. Šis modelis naudoja vienodus svorius vykdant tuo pačiu metu ir panaudojant du skirtingus įvesties vektorius apskaičiuojant palyginamus išvesties vektorius. Šie vienodi tinklai su skirtingais įvesties duomenimis yra sujungti pagal energijos funkciją E . Ši matematinė funkcija apdoroja kai kurias metrikas tarp labiausiai išryškintų bruožų vaizdavimų kiekvienoje pusėje. Parametrai tarp šių vienodų tinkle yra apriboti, kad būtų vienodi. Svių apjungimas užtikrina, kad du labai panašūs vaizdai nebūtų tarp jų atitinkamų tinkle išvesti labai skirtinguose erdvės lokacijose. Taip pat, reikia paminėti, kad tinklas yra simetriškas, nesvarbu kuriam iš vienodų tinklų įvesime vaizdą, visada gausime tokias pačias metrikas. Standartinė išlaidų funkcija skirta mokymosi pavyzdžiui (x_1, x_2) yra pasiūlyta Hadselio. $L(W, (Y, x_1, x_2)) = \frac{1}{2} (1 - Y)(D(W, x_1, x_2))^2 + \frac{1}{2} Y \max(0, m - D(W, x_1, x_2))^2$ (4) where $D(W, x_1, x_2) = \|f(W, x_1) - f(W, x_2)\|_2$. $Y = 1$ if (x_1, x_2) yra panašios poros, ir $Y = 0$ kitu atveju. m yra riba nurodanti norimą slenkstį atstumui tarp x_1 ir x_2 jeigu jie nėra panašūs. Dėl laisvo sureguliuojimo, dažniausiai taip yra sunkiau ištrenuoti trjetų tinklus negu Siamese tinkle.



3 pav. Siamese tinklų modelis

1.4. Nefiksuoto dydžio vaizdai konvoliuciniuose tinkluose

Gilieji konvoliuciniai tinklai reikalauja fiksuoto dydžio įvesties vaizdų. Tačiau realiaame gyvenime nuotraukų ar paveikslukų dydžiai nėra fiksuoti ir varijuoja plačiai. Jeigu yra bandoma primiktynai pakeisti į reikiama dydį, nuotrauką karpant ar deformuojant, informacija, patalpinta nuotraukose bus prarasta. To pasekoje tikslumas nuotraukų klasifikacijos ar objektų identifikavime bus sumažintas ir nepataisomai sugadintas. Nors neuroniniuose tinkluose, konvoliuciniai sluoksniai nereikalauja fiksuoto dydžio įvesties ir geba generuoti specifinius bruožus vaizdo iš bet kokio dydžio nuotraukų. Visgi, pilnai sujungti sluoksniai privalo turėti fiksuoto dydžio įvestį dėl jų pačių apibrėžimo. Dėl to apribojimas fiksuoto dydžio nuotraukų ateina tik iš pilnai sujungtų sluoksnių reikiamos ypatybės. Viena iš šios problemos sprendimo būdų yra naudoti erdvinės piramidės talpinimą.

The spatial pyramid pooling layer extracts the image features from the feature map through the 4×4 , 2×2 , and 1×1 square grid. Then, the SPP layer produces $21 = 16 + 4 + 1$ different spatial bins and obtain a fixed size output by pooling each block. After the spatial pyramid pooling, any feature map can generate 5376-dimensional feature vector, where $5376 = 21 \times 256$. The SPP layer pools the features and generates fixed length outputs, which are then fed into the fully-connected layers. Therefore, the convolution neural network with spatial

2. Įvertinimo funkcija

Kelios įvertinimų metrikos yra naudojamos: panašumo tikslumas bei *score-at-top-K*, kai $K = 30$. Panašumo tikslumas yra išreiškiamas procentaliai pagal tai, kiek trejetų buvo korektiškai sureitinguota. Sakykime, kad turime trejetą, su šiais įvesties parametrais $tt = (x^a, x^p, x^n)$, kur x^p turėtų būti arčiau(panašesnis)šalia x^a . Laikant, kad x^a yra įvesties užklausa žiūrime į rezultatus kitų įvesties duomenų. Jeigu x^p yra reitinguojamas aukščiau nei x^n , tai tada teigiame, kad atitinkamas trejetas yra sureitinguotas teisingai. Kita mums reikalinga metrika, kuri buvo užsiminta anksčiau *score-at-top-K*. Jis nusako skaičių teisingai sureitinguotų trejetų bei atimant iš šio skaičiaus, kuris nusako neteisingai sureitinguotus trejetus iš pogrupio trejetų, kurių reitingas yra didesnis, nei kintamasis K . Pogrupis yra pasirenkamas tokia tvarka: kiekvienam užklauskos paveikslėliui iš duomenų aibės, ištraukia 1000 naujų paveikslėlių iš tos pačios teksto užklauskos ir yra bandoma taip reitinguoti juos, pasitelkiant išmoktas metrikas. Trejeto reitingas yra aukštesnis negu K , jeigu jo x^p arba x^n yra tarp geriausiai reitinguojamų K kaimynų iš užklauskos su paveikslėlių x^a .

2.1. Treniruojami duomenys

Kadangi darbo tema susijusi su detalių vaizdų panašumu, kuris negali būti charakterizuotas pagal paveikslėlių žymes, buvo panaudota trejetų duomenų rinkinys įvertinant vaizdo panašumo modeliams. Buvo paimta 1000 populiariausių teksto užklausų su atrinktais trejetais (x^a, x^p, x^n) su Google 50 paieškos rezultatų kiekvienai užklausa.

Efektyvumas yra nurodomas pirmoje lentelėje. "DeepRanking" parodytas lentelėje yra gilusis reitingavimo modelis treniruotas su 20

Method	Precision	Score-30
ConvNet	82.8%	5772
Single-scale Ranking	84.6%	6245
OASIS on Single-scale Ranking	82.5%	6263
Single-Scale & Visual Feature	84.1%	6765
DeepRanking	85.7%	7004

4 pav. panašumo tikslumo metrikų rezultatai

Method	Precision	Score-30
Wavelet [9]	62.2%	2735
Color	62.3%	2935
SIFT-like [17]	65.5%	2863
Fisher [20]	67.2%	3064
HOG [4]	68.4%	3099
SPMKtexton1024max [16]	66.5%	3556
L1HashKPCA [14]	76.2%	6356
OASIS [3]	79.2%	6813
Golden Features	80.3%	7165
DeepRanking	85.7%	7004

5 pav. Panašumo tikslumo metrikų rezultatai

3. Trejetų tinklo vaizdų atpažinimo tyrimas ir vertinimas

3.1. Kaip išspręsti semantines vaizdų skirtumo problemas

Norint pagerinti atsirandančias vaizdų problemas, aprašytas viršuje

3.2. Treniravimo ir testavimo aplinka

Trejetų tinklo modelis buvo treniruojamas naudojant Lenovo Y-700 kompiuterį su šiomis specifikacijomis: CPU i7 6700K (4 branduolių), 8GB RAM, GPU Nvidia GTX 960m (4GB RAM). Norint pasileisti trejetų modelį būtent šiam kompiuteriui (priklausomai nuo mašinos tai gali skirtis) buvo atsisiųsta Nvidia CUDA programinė įranga su tikslu, kad modelio treniravimas būtų vykdomas pasitelkiant vaizdo plokštę, o ne procesorių. Priežastis kodėl buvo pasirinkta vaizdo plokštė efektyvumas geresnis, nes sugebama ištraukti didesnio dydžio atmintį. Pasirenktas trejetų tinklų modelis yra implementuotas naudojant Tensorflow karkasą, Python 3.5.1. Vaizdų duomenų rinkinyje buvo 3884 nuotraukos, kuriose jau buvo surikiuotos teisinga eilės tvarka, kur pirma nuotrauka pagrindinė, antra nuotrauka - teigiama (kuri yra panaši į pagrindinę) bei neigiama (nepanaši į pagrindinę nuotrauką) ir tokia eilės tvarka yra išsidėstę visos kitos nuotraukos. Ištreniravus šiuos duomenis yra bandoma analizuoti modelio tikslumą imant vaizdų pavyzdžius iš testinio duomenų rinkinio.

3.3. VGG16 tinklo modelis

Panaudota architektūra naudoja VGG16 tinklo bazės sluoksnius klasifikavimui todėl tik labai panašūs vaizdai gali būti naudojami duomenų treniravimui tam, kad pagerinti tinklo atlikimą bei išsaugoti skaičiavimo išteklius. VGG16 yra konvoliucinių neuroninių tinklų modelis, išrastas K. Simonian ir A. Zisserman (Oksfordo universitetas). Modelis sugeba pasiekti 92.7% top-5 testų tikslumą (treniravimo duomenys pasiimti iš ImageNet). VGG16 buvo treniruojamas savaitę iš savaitės naudojant Nvidia Titan Black vaizdo plokščių šeimas.

3.4. Tyrimo rezultatai

Šiame darbe buvo bandyta atpažinti detalius vaizdus, naudojant vaizdų duomenų rinkinį CUHK01. Šiame duomenų rinkinyje yra pavaizduota 3884 viešose vietose einančių pėsčiųjų kadrai. Todėl naudojant šį duomenų rinkinį buvo analizuojama, kaip su pasirinktu trejetų modeliu seksis atpažinti skirtingus žmogaus kadrus, kurie buvo nufotografuoti įvairiai - iš kelių pusių, iš nugaros ir priekio, esant skirtingai žmogaus eisenos pozicijai, skirtingam apšvietimui ir kontrastams, įsiterpiant kitiems objektams prie pėsčiojo. Ištreniravus trejetų tinklą su pasirinktu duomenų rinkiniu ir pradėjus testuoti jo rezultatus, gavau

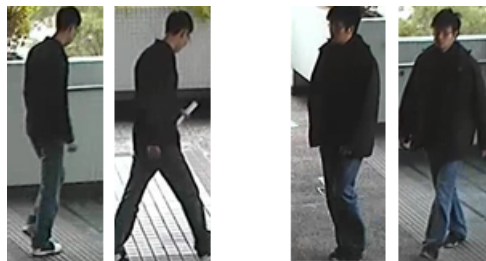
85,7% tikslumo rezultatą. Tai yra gan aukštas įvertinimas, nes atpažinti to paties pėsčiojo skirtingus kadras nėra lengviausia užduotis, nes susiduriame su kliūtimis, aprašytomis anksčiau.



6 pav. Žmogaus skirtingi eisenos kadrai

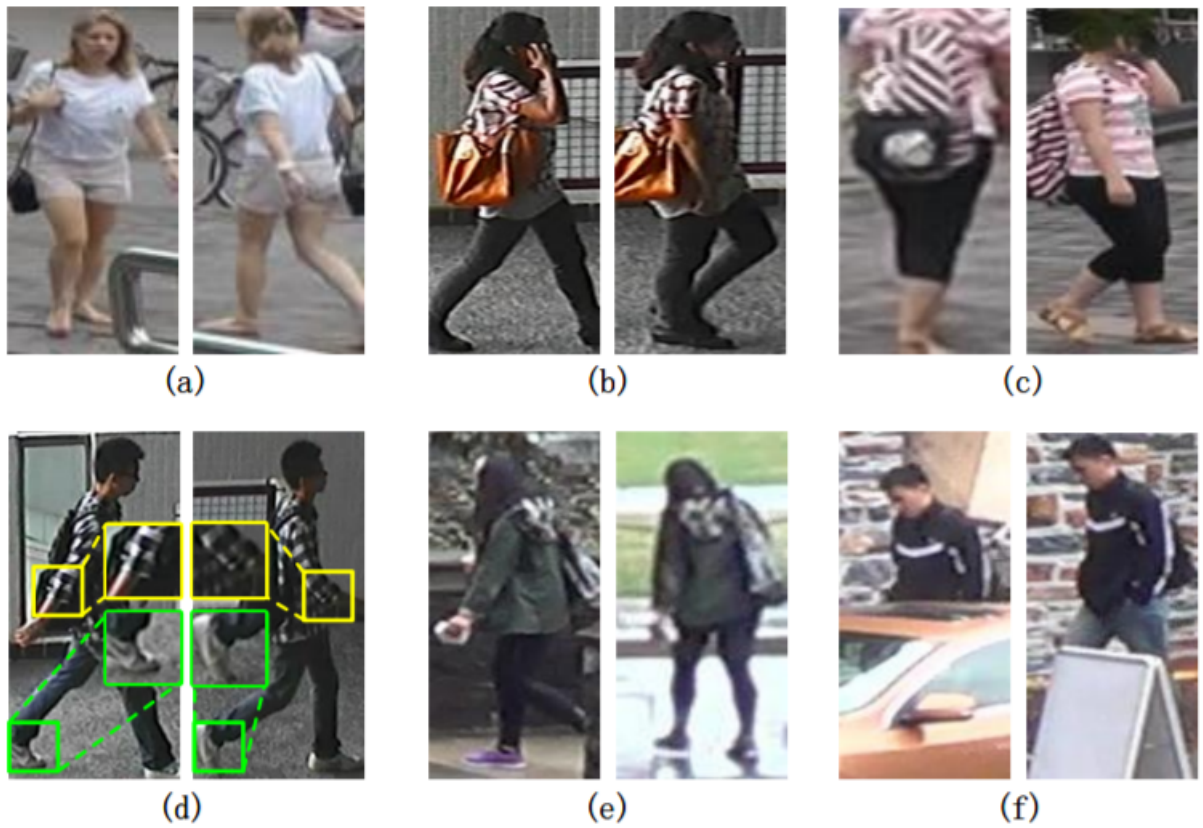
3.5. Vaizdai su prastais rezultatų palyginimais

Visgi ne visi vaizdai buvo tisingai palyginimi ir nustatomi trejetų tinklo modelio. Todėl šiame darbe buvo ieškoma ir nagrinėjama, kurios vaizdų įvestis turėdavo prastus rezultatus bei buvo bandoma išsiaiškinti kas sukeldavo prastus rezultatus.



7 pav. Blogai identifikuotos vaizdų poros

Dauguma nuotraukų, kurios buvo blogai identifikuotos skirėsi viena nuo kitos spalvų intensivumu, skirtingomis spalvomis. Pavyzdžiui matome, kad tiek pirmoje ir antroje nuotraukų porose įsiterpusi nauja žalia spalva (žalumos objektai) leidžia manyti, kad tai yra viena iš priežasčių, kodėl nepavyko tinkamai identifikuoti joms. Taip pat antroje poroje matome, kad žmogaus judėjimo kampas į objektyvą skiriasi, o ir nuotraukų šviesos intensyvumas yra kitoks. Todėl tai leidžia daryti prielaidą, kad vaizdų palyginimo rezultatai bus prasti, jei nuotraukos bus nufotografuotos ne kokybiškai - apšvietimo skirtumai, netvarkingas fonas ar atsiradusi okliuzija.



8 pav. Sunkiai identifikuojamų nuotraukų poros

Viršuje esančios žmonių nuotraukų poros yra sudėtingai identifikuojamos dėl įvairių priežasčių: a) dėl skirtingo objektyvo kampo, b) skirtingos žmogaus pozos, c) prastos nuotraukos kadro užfiksavimo (susiliejas vaizdas), d) ne vienodame lygyje esančios žmogaus kūno dalis, atsiradusios dėl jo judėjimo, e) netvarkingas fonas, f) okliuzija.

3.6. Sprendimo būdai

Visgi ieškant literatūroje informacijos, kurioje būtų aprašyti sprendimo būdai, kai lyginami vaizdai vienas su kitu yra nekorektiški bruožų atžvilgiu, tačiau panašūs ar priklauso tai pačiai klasei, yra siūlomi keli sprendimų būdai. Pavyzdžiui, ypatybių fokusavimą nukreipiant į lokalias detales buvo sukurta nesudėtingas suskirstymas asmens nuotraukos į kelias fiksuotas ir nekintančias dalis, pavyzdžiui į horizontales juosteles ir taip mokantis lokalių nuotraukų bruožų atpažinimo. Visgi, kaip tyrimai iš literatūros nurodė toks skirstymas prastai lygina identifikuojamo asmens kūno dalis. Kituose tyrimuose ir bandymuose buvo bandoma naudoti žmogaus pozą, tokiu būdu identifikuojant skirtingas kūno dalis: kojas, rankas, veidą. Tačiau visgi ir toks atpažinimo metodas yra prastas, kad gauti norimus rezultatus, nes identifikuoti tas pačias žmogaus kūno dalis iš skirtingų žmogaus padėčių kelia problemų dėl jo skirtingos pozos nuotraukose.

4. Rezultatai ir išvados

4.1. Rezultatai

1. Trejetų tinklai palyginti su Siamese modeliu, išskirti trūkumai ir privalumai.
2. Išskirti esamų tyrimų trūkumai
3. Išskirti esamų tyrimų trūkumai.
4. Aprašyti trejetų tinklų pasirinktas architektūros modelis.

4.2. Išvados

1. Yra sudėtinga lyginti trejetų tinklų ir Siamese modelius, nes jų architektūra gali varijuota. Teisingiau yra lyginti specialius jų sukurtus modelius vienas su kitu.
2. Pasirinktas trejetų tinklų modelis parodė neblogus rezultatus lyginant žmonių eisenos kadrus.
3. Naudojant vaizdo plokštę, o ne procesoriaus atmintį, ištreniruoti galima žymiai greičiau.

4.3. Rekomendacijos ateities darbams

1. Atlikti Cassandra ir Hyperledger Fabric praeinamumo ir vėlavimo tyrimą atlikti naudojant tą pačią kompiuterinę įrangą.
2. Atlikti Cassandra praeinamumus ir vėlavimo tyrimą keičiant konfigūraciją link transakcijų neprieštaravimo pusės ir lyginti lėtėjantį greitį su Hyperledger Fabric greičiu.
3. Atlikti greičio lyginimo tyrimus su skirtingomis Fabric ir Cassandra versijomis.

5. Priedai

5.1. Žodynas

- Svoris(angl. weight) - parametras neuroniname tinkle, kuris transformuoja įvesties duomenys paslėptuose sluoksniuose.
- Nuostolių funkcija(angl. loss function) - funkcija, padedanti optimizuoti svorius, taip sumažinant neatitikimų nuostolius.
- Detalieji vaizdai(angl. fine-grained) - detalūs vaizdai. Vaizdo klasifikavimo užduotyse, tai yra įvesties vaizdai, kuriuos yra sudėtinga išskirti klasėms, pvz.: identifikuojant skirtingų markių automobilius.
- Aktyvacijos funkcija(angl. activation function) - funkcija, skirta nustatyti ar neuronas turi būti aktyvuotas, skaičiuojant svorių sumą bei pridedant postūmio parametą.
- Postūmis(angl. bias) - papildomas parametras neuroniniuose tinkluose, kuris padeda koreguoti išvesti kartu su svorių įvesties suma, skirta neuronams perduoti. Taip pat šis parametras leidžia perstumti aktyvacijos sumą nuo kairės į dešinę.

Literatūra

- [BGL⁺90] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Sickinger ir Roopak Shah. Signature verification using a siamese time delay neural network, 1990.
- [JYT⁺14] Wang Jiang, Song Yang, Leung Thomas ir Rosenberg Chuck. Learning fine-grained image similarity with deep ranking, in cvpr, 2014.
- [MKR16] Iaroslav Melekhov, Juho Kannala ir Esa Rahtu. Siamese network features for image matching, 2016.