

VILNIAUS UNIVERSITETAS
INFORMATIKOS INSTITUTAS
PROGRAMŲ SISTEMŲ KATEDRA

Baigiamasis bakalauro darbas

Neuroninio tinklo atsparumas transformacijoms
(Neural Network Resistance to Transformations)

Atliko: Lukas Povilonis (parašas)

Darbo vadovas: dr. Vytautas Valaitis (parašas)

Darbo recenzentas: dr. Julius Andrikonis (parašas)

Vilnius
2020

Santrauka lietuvių kalba

Viena iš plačiai studijuojamų kompiuterinės vizijos problemų yra panašumų ir skirtumų atradimas tarp dviejų paveikslėlių. Šioje srityje taip pat yra naudojami ir neuroniniai tinklai, kurie yra apmokami dideliais kiekiais duomenų. Neuroniniai tinklai šią problemą gali išspręsti greičiau lyginant su kitais problemos sprendimo būdais.

Nepilotuojamoms oro transporto priemonėms (dronuose) susiduria su vietovės nustatymo problema palydovų signalų nepasiekiamose vietose. Šiai problemai spręsti gali būti pasitelktos dronuose esančios kameros kurių daromos nuotraukos yra lyginamos su topografiniu žemėlapiu pasitelkus apmokytą neuroninį tinklą. Padarytos nuotraukos gali nesutapti su topografiniu žemėlapiu, nes jos bus iškraipytos fotoaparato lęšiu ir fotografavimo kampo. Todėl šiame darbe yra simuliuojamos galimos foto nuotraukos distorsijos ir atliekamas distorsijų poveikio tyrimas apmokytam neuroninio tinklo modeliui.

Summary

One of the widely studied problems of computer vision is the discovery of similarities and differences between the two images. Neural networks, which is trained by large amounts of data, are also used in this area. Neural networks can solve this problem faster compared to other solutions.

Unmanned aerial vehicles (drones) face the problem of positioning in places inaccessible to satellite signals. To solve this problem, cameras in drones can be used to take photos of which are compared with a topographic map using a trained neural network. The photos taken may not match the topographic map as they will be distorted by the camera lens and the shooting angle. Therefore, in this thesis, possible image distortions are simulated and a study of the effects of distortions on a trained neural network model is performed.

Turinys

Santrauka lietuvių kalba	2
Summary	3
Įvadas	5
4. Dirbtiniai neuroniniai tinklai	6
5. Aktyvacijos funkcijos	7
6. Vienasluoksniai ir daugiasluoksniai neuroniniai tinklai	8
7. Neuroninių tinklų mokymas	9
4. Neuroninių tinklų bibliotekos	10
4.1. Tensorflow	10
4.2. Keras	11
4.3 OpenCv	11
5. Naudojamas neuroninis tinklo modelis	11
6. Trejetų panašumas naudojant neuroninį tinklą	13
7. Tranformacijos	14
7.1. Afjininė tranformacija	14
7.2. Projekcinė tranformacija	15
7.3. Suliejimo transformacija	16
7.4. Statinės ir pagalvės formos distorsiją	16
8. Neuroninio tinklo atsparumas transformacijoms	18
8.1. Suformuluota užduotis	18
8.2. Pasirinktos užduoties sprendimas	18
8.2.1. Tyrime naudojami įrankiai	19
8.3. Tyrimas	20
8.3.1 Afjininės transformacijos rezultatai	20
8.3.2 Projekcinė transformacijos rezultatai	21
8.3.3 Suliejimo transformacijos rezultatai	23
8.3.4 Statinės formos transformacijos rezultatai	24
8.3.5 Pagalvės formos transformacijos rezultatai	25
Rezultatai ir išvados	27
Šaltiniai	28

Ivadas

Dirbtiniai neuroniniai tinklai yra lyg netiksliai imituoti kai kurių gyvųjų organizmų informaciniai procesai, vykstantys jų smegenyse. Jie pritaikomi įvairiose srityse grafinių vaizdų atpažinime, realių vaizdų atpažinime, garso atpažinime, finansinėse sistemose, teksto atpažinime [GV07]. Šie neuroniniai tinklai skaldo duomenis į dalis ir naudodami statistines funkcijų rinkinius sugeba juos efektyviau sulyginti. Iš šių lyginimų dirbtinis neuroninis tinklas sugeba mokytis gaunamų duomenų dėsningumu ir atpažinti juos kituose panašiuose objektuose. Jeigu vis daugiau atliekant šiuos veiksmus kompiuteris juos atlieka tiksliau yra teigiama jog kompiuteris mokosi. Duomenų kiekiams vis didėjant ir užduotims vis sunkėjant pritaikytai kompiuterinei sistemai gali užimti ganėtinai daug laiko. Žinoma dar daugiau laiko užima atliekant užduotis su vaizdu.

Surasti panašumus ir skirtumus tarp dviejų paveikslėlių yra viena iš plačiai studijuojamų problemų kompiuterinės vizijos (angl. *computer vision*), nes ši problema iškyla daugelyje sričių. Viena iš tokių yra bepiločių orlaivių pilotavimas zonose, kai neveikia globalios padėties nustatymo sistema (GPS). Šiai problemai spręsti gali būti pasitelktos dronuose esančios kameros kurių daromos nuotraukos yra lyginamos su topografiniu žemėlapiu pasitelkus apmokytą neuroninį tinklą.

Šio darbo tikslas — ištirti apmokyto neuroninio tinklo modelio atsparumą transformacijoms, kurios gali atsirasti fotografuojant objektus.

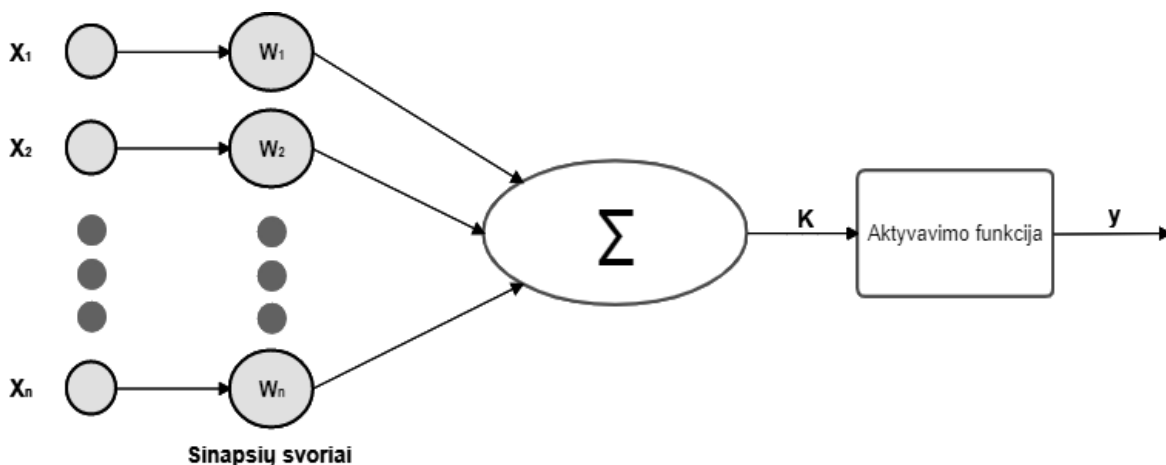
Darbo uždaviniai:

1. Teoriškai aprašyti problemą ir paruošti duomenis.
2. Išreikšti transformacijas matematiškai.
3. Atlikti praktinį bandymą su apmokyto neuroninio tinklo atsparumu transformacijoms

4. Dirbtiniai neuroniniai tinklai

Dirbtinis neuroninis tinklas (toliau DNT) dažniausiai vadinamas neuroniniu tinklu, jo atsiradimo priežastis buvo atrastas skirtumas jog žmogaus smegenys atlieka skaičiavimus visiškai kitokiu būdu nei įprastas skaitmeninis kompiuteris. Smegenys yra labai sudėtingas, nelinijinis ir lygiagretusis kompiuteris (angl. parallel computer). Jos turi gebėjimą sisteminti struktūrinės sudedamąsias dalis, žinomas kaip neuronai, tokiu būdu, kad atliktų tam tikrus skaičiavimus (pvz., modelio atpažinimas, suvokimas ir motorikos valdymas). Žmogaus regėjimas yra informacijos apdorojimo užduotis, tai vizualinės sistemos funkcija, kuri leidžia apibūdinti aplinką ir orientuotis aplinkoje. Smegenys paprastai atlieka suvokimo atpažinimo užduotis (pvz., Atpažįsta pažystamus veidus, nepažįstamoje vietoje) maždaug per 100-200 ms, o daug mažesnio sudėtingumo užduotis skaitmeniniai kompiuteriai atlikinėja ilgiau. Neuroninis tinklas yra mašina, kuri yra sukurta smegenų veikimo modeliu, koku smegenys atlieka tam tikras užduotis [HKC15].

DTN sudaro tarpusavyje sujungtų dirbtinių neuronų grupę. Dirbtinis neuronas, tai biologinio neurono abstrakcija, pagrindinis dirbtinių neuroninių tinklų komponentas. Pirmoje schemoje pavaizduota neurono modelis, kuris formuoja pagrindą, kurti didelę neuronų tinklų šeimą. Trys pagrindiniai neurono modelio elementai, tai svoriai, slenksčiai ir aktyvavimo funkcija. Svorio koeficientai - $W = [W_1, W_2, \dots, W_n]$. Šis koeficientas rodo atskirų įvesčių stiprumą ir yra aprašomas vektoriumi $X = [x_1, x_2, x_3, \dots, x_n]$. Visų įvesčių signalai yra dauginami iš svorio koeficiento ir gaunama neuroninė jungtis XW . Pagrindinė taisyklė, kad XW sužadintų signalą išvestyje y yra teigiamas svorio koeficientas, o jei jis yra neigiamas – tada XW slopina išvesties signalą [WSL13].



1 pav. Dirbtinis neuronas

Šis signalas po to keliauja į aktyvacijos funkcija, kuri nusako ar susumuotas svoris bus aktyvuotas ar ne.

5. Aktyvacijos funkcijos

Pateikiama aktyvavimo funkcija, kuri aprašo kaip vidinis neurono slenkstis veikia neurono y išvesties aktyvavimą:

$$y = f(\sum_{i=1}^n XW \theta). \quad (1)$$

Matematinės operacijos su išvesties signalu vadinamos aktyvavimo funkcija. Nuo sprendžiamos problemos arba uždavinio priklauso aktyvavimo funkcijos sudėtingumas. Populiariausios funkcijos yra tiesinė, slenksčio, Piecewise'o tiesinė, hiperbolinė tangento ir sigmoidinė aktyvavimo .

Tiesinė funkcija: naudojant tiesinę aktyvavimo funkciją gautas parametras iš neuroninio tinklo dauginamas iš konstantos c , tai vaizduoja formulė (2)

$$\hat{y} = f(K). \quad (2)$$

Slenksčio funkcija: (*angl. Binary step*) naudojant slenksčio formulę parametras gautasis iš neuroninio tinklo yra lyginamas su pasirinkta konstanta c . Kai gautasis neuroninio tinklo parametras K yra didesnis arba lygus pasirinktai konstantai c , tai aktyvavimo formulė gražina parametą 1. O kai parametras K yra mažesnis už pasirinktą konstantą c , tai aktyvavimo funkcija gražina 0. Tai yra pavaizduota formulėje trečioje

$$f(K) = \begin{cases} 0, & \text{kai } K < c \\ 1, & \text{kai } K \geq c \end{cases}. \quad (3)$$

Sigmoidinė funkcija: ši funkcija dar vadinama logistinė. Sigmoidinė funkcija dažniausiai naudojama modeliuose kuriuose norima kažką, tai nuspėti, nustatyti tikimybę. Funkcija naudojama todėl jog ji egzistuoja tarp režių 0 ir 1 [SSAF17].

$$f(K) = \frac{1}{1 + e^{-K}}. \quad (4)$$

Hiperbolinio tangento funkcija: kaip ir sigmoidinė funkcija ši tangento funkcija yra loginė funkcija kuri naudojama norint nustatyti tikimybę. Jos pagrindinis skirtumas nuo sigmoidinės funkcijos yra tai jog šios funkcijos režiai yra nuo -1 iki 1.

$$f(K) = \tanh(K) = \frac{(e^K - e^{-K})}{(e^K + e^{-K})}. \quad (5)$$

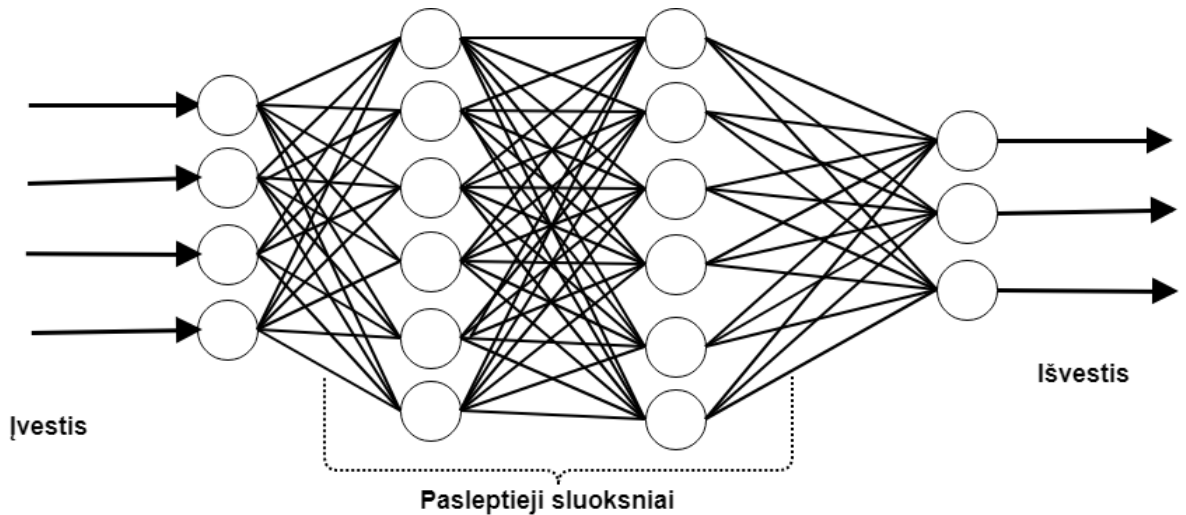
Piecewise tiesinė funkcija: ši funkcija yra hibridinė funkcija tarp slenkstinės funkcijos ir hiperbolinės tangento formulės. Ši formulė nenutrūksta taip kaip tai daro slenksčio funkcija ir savo pajėgumu nieko nenusileidžia hibridiniai tangento funkcijai [ANPLU18].

$$f(K) = \max(\alpha(K + c) - c, \min(\alpha(K - c) + c, K)), \quad (6)$$

$$f'(K) = \begin{cases} \alpha, & \text{kai } |K| > 0 \\ 1, & \text{kai } |K| < 0 \end{cases}. \quad (7)$$

6. Vienasluoksniai ir daugiasluoksniai neuroniniai tinklai

Neuroniniame tinkle neuronai dažniausiai yra grupuojami į sluoksnius, pagal tai jei skirstomi į daugiasluoksnius ir viena sluoksnius. Neuroniniai tinklai, kurie grupuojami į sluoksnius turi įėjimo sluoksnį, paslėptą sluoksnį ir išėjimo sluoksnį. Įėjimo sluoksnyje yra įvedamos kintamųjų vertės. Paslėptajame sluoksnyje yra apdorojama informacija gaunama iš įvesties sluoksnio. Išėjimo sluoksnyje neuronai gauna informaciją ir ją išsiunčia iš neuroninio tinklo.



2 pav. Daugiasluoksnis neuroninis tinklas

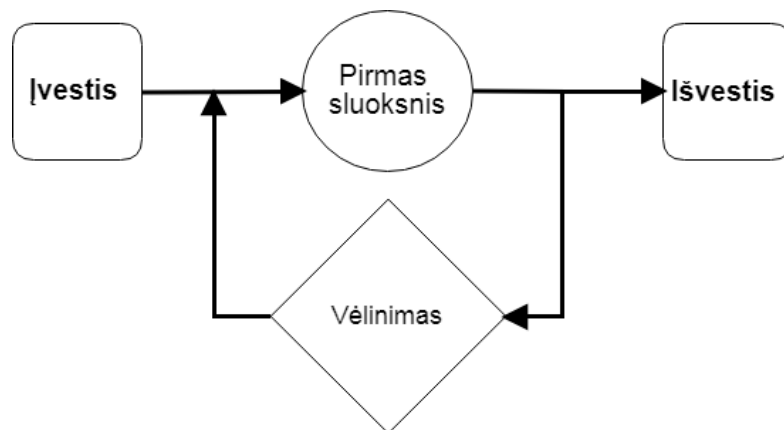
Sudėtingoms užduotims spręsti naudojami tinklai turintys daugiau nei vieną svorių sluoksnį. Neuroniniuose tinkluose dažnai sunku nustatyti sluoksnių ir elementų skaičių. Didėjant neuronų skaičiui ir sluoksnių skaičiui tinklas darosi sudėtingesnis. Reikalingas neuronų skaičius priklauso ne nuo įėjimo erdvės matavimų skaičiaus, o nuo pačio uždavinio komplikotumo. Daugiasluoksniai neuroniniai tinklai pasižymi tuo, kad dažniausiai turi kur kas daugiau adaptuojamų sluoksnių nei viena sluoksnius [LYZ17].

Priklausomai nuo signalų siuntimo krypties skiriami į vienkrypčius (tiesioginio ryšio) ir grįžtamojo ryšio. Vienkrypčio ryšio neuroninių tinklų vieno sluoksnio išvestys jungiasi tik su kito sluoksnio išvestimi. Ryšiai nėra įmanomi tarp vieno sluoksnio išvesčių ir to paties ar prieš tai einančio sluoksnio įvesčių.



3 pav. Tiesioginio ryšio neuroninis tinklas

Grįžtamojo ryšio neuroniniuose tinkluose mokymo imtis praeina pro visą neuroninį tinklą. Šio tinklo įvestis yra sudaryta iš išorinės įvesties ir tinklo išvesties. Gauti išėjimo rezultatai yra nukreipinėjami į įvestį tol kol rezultatai tampa stabilūs. Schemoje pateikiamas grįžtamo ryšio tinklo modelis [LYZ17].



4 pav. Grįžtamojo ryšio neuroninis tinklas

7. Neuroninių tinklų mokymas

Neuroninio tinklo kūrimo pagrindas yra mokymas arba adaptacija. Atliekant neuroninio tinklo mokymą norima įvertinti didžiausią tikimybę turintį statistinį proceso modelį, kuris kuria mokymo duomenis. Neuroninio tinklo savybė generuoti išvadas ir atlikti tinkamą naujų duomenų įvertinimą yra vadinama bendrinimo savybė [BK17].

Proceso pradžioje parametrai būna netinkami ir sistemoje atsiranda didelių klaidų. Mokymo metu parametrai derinami taip, kad klaidų skaičius sistemoje mažėtų. Pati sistema atlikdama „susipažinimą“ su duomenimis atrandą patį tinkamiausią parametrų rinkinį. Adaptyviosios sistemos pasižymi savo sudėtingumu. Šios sistemos privalo vykdyti reikalingas funkcijas ir be to jos naudojamos posistemių parametrų keitimui. Atsiradus duomenų pasikeitimui ši metodika leidžia modifikuoti sistemos parametrus surasti geriausią uždavinio sprendimą [GV07].

Priklausomai nuo neuroninių tinklų tipų yra naudojami skirtingi mokymo metodai. Patys pagrindiniai DTN apmokymų tipai yra mokymas su mokytoju ir mokymas be mokytojo. Egzistuoja ir trečiasis neuroninių tinklų mokymo tipas, tai skatinantis mokymas.

Mokymo su mokytoju atveju dalis duomenų yra sužymėti, tai leidžia surasti naujų duomenų prognozavimo taisykles. Šio mokymo problemos turi kvalifikaciją ir regresiją. Mokymas be mokytojo atveju duomenys yra nesužymėti, todėl atliekami tik grupavimo arba taisyklių atradimo veiksmai. Abu veiksmai yra naudojami skirtingam rezultatui gauti (pvz. Grupavimas – naudojamas pirkėjų įpročių atpažinimui) [GV07].

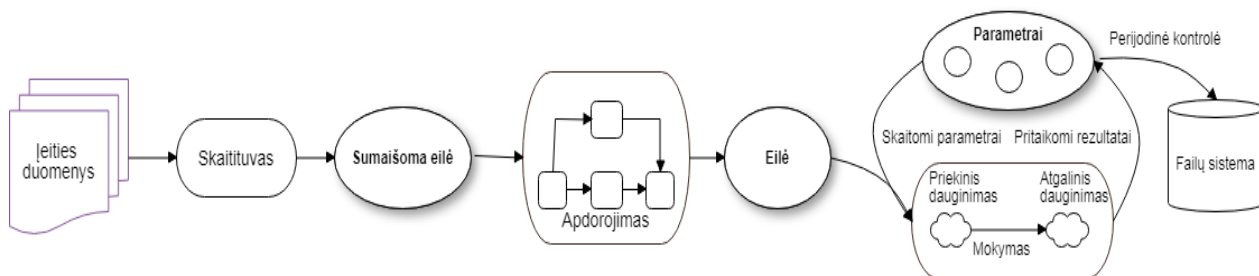
Atliekant neuroninių tinklų mokymus yra atrasta daug taisyklių. Viena iš seniausių taikomų taisyklių yra Hebbo taisyklė. Daugelis naujų taisyklių paremtos Hebbo taisyklės principais. Taip pat naudojama Grosbergo funkcija, Delta taisyklė, Cauchy mašina ir daugelis kitų.

4. Neuroninių tinklų bibliotekos

Kadangi šiuo momentu neuronis tinkas vis daugiau naudojamas industrijose ir moksliniuose tyrimuose todėl yra didelis pasirinkimas neuroninių tinklų bibliotekų. Daugelio jų naudojimas yra nesudėtingas. Daugelis bibliotekų išnaudoja didelę techninės įrangos greitumą. Šias bibliotekas kuria tokios įmonės kaip: Microsoft, Google, Facebook ir kitos didelės korporacijos.

4.1. Tensorflow

Tensorflow – neuroninių tinklų biblioteka naudojama greitam neuroninių tinklų prototipavimui. Ši programinė įranga gali veikti su daugeliu skaičiavimo (angl. *computational*) prietaisų tarp kurių yra keletą branduolių turinčių procesorių (angl. *Multicore CPUs*) ir paprastų vaizdo plokščių (angl. *generalpurpose GPU*). Šią sistemą galima naudoti labai lanksčiai išreiškiant įvairius algoritmus. Naudojama neuroninių tinklų sistemoms kurios yra naudojamos daugelyje inžinerinės informatikos sričių ir daugeliu kitų mašininio mokymosi (angl. *machine learning*) sričių tarp, kurių yra ir balso atpažinimo, vaizdo atpažinimo (angl. *computer vision*). Penktame paveikslukas vaizduoja paprastą apmokymą su keliais subgrafais (angl. *subgraphs*), kurie veikia konkurencingai ir sąveikauja per bendrai naudojamus parametrus ir eiles (angl. *queues*). [AAA12]



5 pav. Tensorflow mokymo procedūros pavyzdys

4.2. Keras

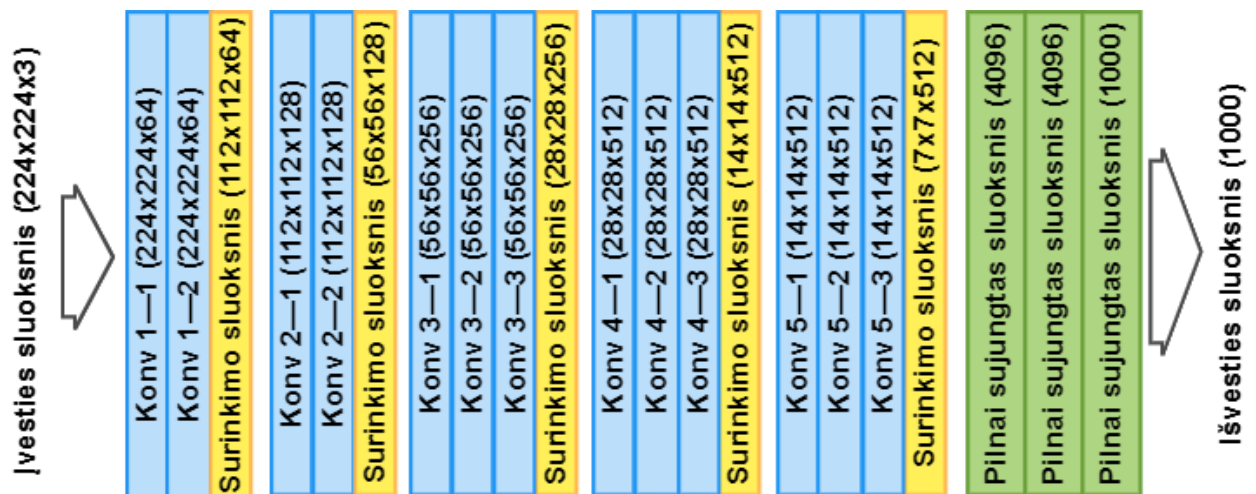
„Keras“ – aukšto lygio neuroninių tinklų aplikacijų programavimo sąsaja (angl. *Application Programming Interface, API*), parašyta „Python“ programavimo kalba ir galinti veikti su „TensorFlow“, „CNTK“. „Keras“ yra sukurta susitelkiant leisti galimybę greitam eksperimentavimui, protatipavimui. Jis palaiko tiek konvoliucinius neuroninius tinklus (angl. *Convolutional networks, CNN*), tiek ir pasikartojančių (angl. *Recurrent networks, RNN*), tiek ir kombinacijoje dviejų neuroninių tinklų. Gali veikti su procesoriumi ar su vaizdo plokšte.

4.3 OpenCv

OpenCV yra atviro kodo kompiuterinės vizijos biblioteka. Šitoje bibliotekoje yra daugiau nei 2500 optimizuotų algoritmų, kurie yra lengvai suprantami ir naudojami klasikiniai ir pažangiausi kompiuterinės vizijos algoritmai. Šią biblioteka galima naudoti norint atpažinti objektus, atskirti žmonių veiksmus vaizdo įrašė, sekti kameros judėjimą ar sujungti du paveikslukus sukuriant didelės raiškos panoraminį paveikslėlį. OpenCV biblioteka turi C++, Python, Java, ir MATLAB sąsajas. Ir palaiko Windows, Linux, Android ir Mac Os operacines sistemas.

5. Naudojamas neuroninis tinklo modelis

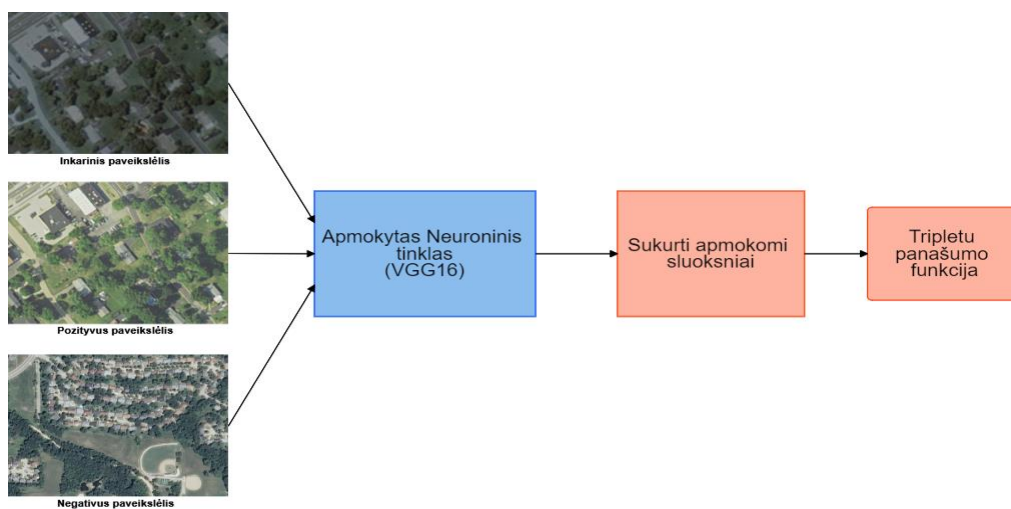
Darbe yra naudojamas modifikuotas VGG 16 konvoliucinis neuroninio tinklo modelis. VGG 16 neuroninio tinklo modelis yra sugalvotas ir pasiūlytas Oksfordo universiteto studentų: K. Simonyan ir A. Zisserman[] VGG 16 vardas kilęs iš to jog jis turi 16 sluoksnių tarp kurių yra: surinkimo sluoksniai (angl. *pooling*), pilnai sujungti sluoksniai (angl. *Dense*), konvoliuciniai sluoksniai. Šis modelis yra pavaizduotas šeštame paveikslėlyje.



6 pav. VGG 16 neuroninio tinklo modelis

Kaip galime suskaičiuoti modelis sudarytas iš 21 sluoksnių: 13 konvulsinių sluoksnių, 5 pilnai surinkimo sluoksnių, 3 pilnai sujunktų sluoksnių, bet tik 16 sluoksnių turi svorių.

Trejetų paremto neuroninio tinklo architektūra yra parodyta septintame paveikslėlyje. Šitoks tinklas trejetus: inkarinį, pozityvų ir neigiamą paveikslėlius apdoroja atskirai į tris vienodus neuroninio tinklo modelius, kurie dalinasi parametrais ir architektūra. Gilus neuroninio tinklo modelis yra sudarytas iš pilno jau apmokyto neuroninio tinklo VGG16 visų sluoksnių kurio pats paskutinis sluoksniis yra pakeistas į naujai sukurtus.

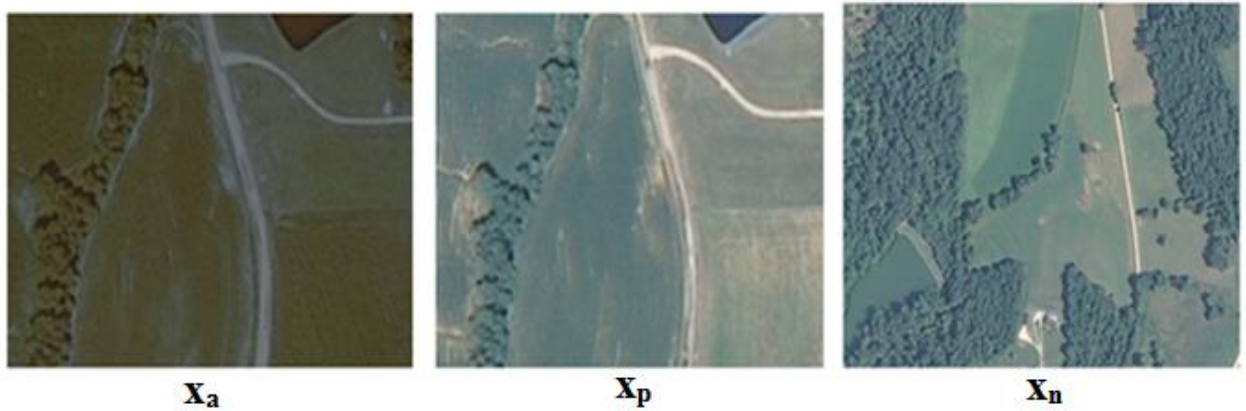


7. pav. Trejetų paremtas neuroninio tinklo modelis

Šio darbo tyrimo dalyje naudojamas neuroninio tinklo modifikuotas VGG 16 modelis. Naudojamas neuroninio tinklo modelis susideda iš viršutinių šešių VGG16 modelio sluoksnių ir pridėtų naujai sukurtų ir pritaikytų sluoksnių. Jam apmokyti naudojama trejetų panašumo formulė 10.

6. Trejetų panašumas naudojant neuroninį tinklą

Trejetai (*angl. Triplet*) tai yra trys objektai dažniausiai paveikslukai, nuotraukos ar žemėlapio vaizdai. Trejetų serijomis yra apmokomas neuroninis tinklas, kuris gali palyginti vaizdus ir įvertinti jų panašumą. Tokiu atveju kalbame apie tokius trejetus, kurie susideda iš trijų paveikslukų: norimo tikrinti (inkarinis) (*angl. Anchor*), kuris aštuntame paveiksliuke žymimas X_a , teigiamo (panašaus) paveiksluko (*angl. Positive*), kuris aštuntame paveiksliuke žymimas X_p ir neigiamo (nepanašaus) (*angl. Negative*) paveiksluko, kuris aštuntame paveiksliuke žymimas X_n .



8 pav. Trejeto pavyzdys

Neuroninio tinklo mokyme siekiame, kad trejetų dalis X_a laikytųsi arčiau X_p , nei prie X_n , tai pavaizduota septintajame paveikslėlyje. Tokios trejetų neuroninio tinklo savybės naudojamos daugelyje sričių: veido atpažinime, vaizdo atkūrimui, žmogaus indentifikavime. Visos šios sritys naudoja trejeto panašumo formulę (*angl. Triplet Loss function*).

Viena labiausiai naudojamų trejetų panašumo formulė yra pavaizduota formulėje numeriu 8.

$$\sum_{i=1}^{N_{tr}} \max(d(f_i^a, f_i^p) - d(f_i^a, f_i^n) + \alpha, 0), \quad (8)$$

Kur raide d žymimas distancijos parametras, kuris yra toks jog $(f_i^a, f_i^p) < d(f_i^a, f_i^n)$ tai reiškia jog $d(f_i^a, f_i^n)$ distancija tarp pozityvaus ir inkarinio paveikslėlio yra mažesnis, nei $d(f_i^a, f_i^n)$ kas yra distancija tarp inkarinio ir negatyvaus paveikslėlio. Funkcijoje N_{tr} yra trejetų kiekis, o alfa yra parametras didesnis už 0, kuris reguliuoja atstumo dydį tarp dviejų paveikslėlių porų.

Kita dažnai naudojama trejetų panašumo funkcija veido atpažinimo uždaviniuose yra formulė 9.

$$\sum_{i=1}^{N_{tr}} [\|f_i^a - f_i^p\|_2^2 - \|f_i^a - f_i^n\|_2^2 + \alpha]_+, \quad (9)$$

ši formulė naudoja Euklidinį atstumą kaip distancijos parametą. Kaip ir prieš tai aprašytoje formulėje šioje formulėje α yra mažas pozityvus skaičius, naudojamas kaip dydis tarp dviejų paveikslėlių porų ir sumuoti absoliučias reikšmes, o f_i^a, f_i^p, f_i^n atitinkamai reiškia inkarinis, pozityvus ir negatyvus paveikslėlis.

Šiame darbe naudojamo neuroninio tinklo modelyje yra naudojama ši modifikuota panašumų formulė[MOA18]:

$$\sum_{i=1}^{N_{tr}} \left[-\ln \left(\frac{\sum_{j=1}^N (f_{ij}^a - f_{ij}^p)^2}{\beta} + 1 + e \right) + \ln \left(\frac{\sum_{j=1}^N (f_{ij}^a - f_{ij}^n)^2}{\beta} + 1 + e \right) \right], \quad (10)$$

čia $f_{ij}^a, f_{ij}^p, f_{ij}^n$ yra neuroninio tinklo inkarinio, panašaus ir nepanašaus paveikslėlio išvestys atitinkamai. N_{tr} yra trejetų skaičius naudojamas neuroninio tinklo apmokymui. N yra išvesties vektoriaus dimensijos, mažas pozityvus skaičius žymimas ženklu e . Panašumo formulė lygi 0, kai pozityvus ir negatyvus paveikslėlis yra nutolę maksimalia distancija.

$$\text{sim}(im_1, im_2) = -\ln \left(-\frac{\sum_{j=1}^N (f_{1j} - f_{2j})^2}{\beta} + 1 + e \right). \quad (11)$$

Su šita panašumo skaičiavimo formule galime skaičiuoti panašumą tarp dviejų paveikslėlių. Formulėje $f_{1j} = f(im_1)_j, f_{2j} = f(im_2)_j$, N yra demencijų skaičius ir $j \in [1..N]$. panašumui nustatyti dar galima naudoti paprastą Euklidinę formulę, bet neuroninio tinklo modelyje yra panaudota modifikuota trejetų panašumo formulė 2, nes ši metrika gauna mažesnius parametrus.

7. Transformacijos

7.1. Affininė transformacija

Afininė transformacija (angl. *Affine transformation*). yra tokia transformacija, kuri išlaiko kolinijškumą (kai visi linijoje esantys taškai išlaiko savo buvimo vietą) ir atstumų santykį (tai yra kai vidurio taškas lieka vidurio taškas po transformacijos). Nors afininė transformacija išlaiko linijų proporcijas, bet tai nereiškia jog ši transformacija išlaiko linijų ilgus ir kampus su kitomis linijomis.



9 pav. Afininės transformacijos pavyzdys

Afininė transformacija gali bet kokį trikampį paversti į bet kokį kitą.

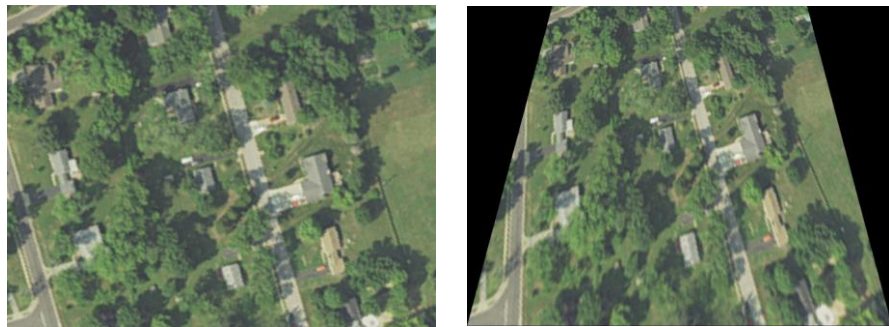
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = s \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} = s \begin{bmatrix} \cos \alpha (x - x_0) + \sin \alpha (y - y_0) \\ -\sin \alpha (x - x_0) + \cos \alpha (y - y_0) \end{bmatrix}, \quad (12)$$

12 formulėje yra parodytas afininė transformacija su vertumu mes vertimų nenaudosime, todėl supaprastinsime šiek tiek formalią 13.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = Ax + B; \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}. \quad (13)$$

7.2. Projekcinė transformacija

Projekcinė transformacija (*angl. Perspective transformation*) tai tokia transformacijai kuri išlaiko tiesias linijas, tiesios linijos išsilaiko net ir po transformacijos. Šioje transformacijoje toliau esantys tokio pat dydžio objektai yra atvaizduojami mažesni, nei arčiau esantys, todėl jai atvaizduojami geležinkelio bėgiai toliau esantys atrodo jog sueina į vieną tašką.



10 pav. Projekcinės transformacijos pavyzdys

Šitai transformacijai apskaičiuoti naudojame homogenines koordinates (*angl. homogeneous coordinates*).

$$\tilde{x}' = \tilde{H}\tilde{x}. \quad (14)$$

Keturioliktoje formulėje $\tilde{H}$ žymima 3×3 matrica. Reikia pažymėti jog $\tilde{H}$ homogeninė matrica todėl, jai matricos skiriasi tik dydžiu jos yra ekvivalentės. Parametras $\tilde{x}$ turi būti normalizuotas norint išlaikyti reikia mus rezultatus [RSCCA10]

$$x' = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + h_{22}} \text{ ir } y' = \frac{h_{10}x + h_{11}y + h_{12}}{h_{20}x + h_{21}y + h_{22}}. \quad (15)$$

7.3. Suliejimo transformacija

Suliejimo transformacija norima imituoti blogai sufokusuotas nuotraukas jai imituoti naudojame OpenCv bibliotekos metodą „resize“, kuris pakeičia paveikslėlio dydį pagal pasirinktus matmenis. Interpoliacijos metodą kuris buvo naudojamas resize metode yra linijinė interpoliacija, šis metodas nusako kokia tu ir būti tam tinkamo pikselio.

$$y = y_a + (y_b - y_a) \frac{x - x_a}{x_b - x_a}. \quad (16)$$

Linijinė interpoliacija paima du duomenų taškus šiuo atveju pikselius ir jai pasirinksime du pikselius tarkim $(x_a - y_a)$ ir $(y_b - y_b)$ ir interpoliuojame pagal 16 formulę. Paveikslėlyje numeriu 11 galima pamatyti kaip atrodo suliejimo transformacija, kaip paveikslėlis yra išlietas ir atrodo blogai sufokusuotas.



11 pav. Statinės transformacijos pavyzdys

7.4. Statinės ir pagalvės formos distorsija

Tai yra optinės distorsijos kurios yra priešingos viena kitai. Statinės distorsija (*angl. Barrel distortion*) kitaip dar vadinama žuvies akies distorsija. Statinės distorsiją galima apibrėžti kaip distorciją kuri mažina atvaizdo atstumą tolystant nuo optinių ašių. Tai reiškia jog tolystant nuo paveikslėlio centro į kurią nors pusę atvaizduojamas atvaizdas paveikslėlyje yra labiau iškraipomas ir neatitinka tikrovės. Šitai distorsijai atvaizduoti naudosime formulę 17 [AFSLM01].

$$r_d = r_u(1 + X|r_u|^2) \leftrightarrow r_u = \frac{r_d}{1 - Xr_d^2}, \quad (17)$$

kur r_d ir r_u yra distancijos parametras, kuris rodo taško nutolimą nuo distorsijos centro. Parametras X yra distorsijos parametras kuris yra specifinis naudojamai optiniam lęšiui. Šiame

darbe naudosime dešinėje pusėje pavaizduotą formulę. paveikslėlyje yra pavaizduotas kaip atrodo statinės transformacija.



12 pav. Statinės transformacijos pavyzdys

Pagalvės formos distorsijai (*angl. Pincushion*) naudosime tą pačią formulę 17, kuri yra naudojama ir statinės formos distorsijai tik naudosime priešingus parametrus. 13 paveikslėlyje yra pavaizduota kaip atrodo ši distorsija.



13 pav. Pagalvės transformacijos pavyzdys

8. Neuroninio tinklo atsparumas transformacijoms

Dažniausiai apmokytas neuroninio tinklo modelis yra naudojamas atpažinti ar atskirti kažkokį tai objektą iš duoto paveikslėlio. Bet kai neuroninis tinklas yra pradėtas naudoti nepilotuojamose lėktuvuose atsiranda tikimybė jog naudojama kamera gali turėti klaidų ar padarytos nuotraukos kampas gali neatitikti naudojamuose duomenų rinkiniuose. Su šiomis padarytomis nuotraukomis neuroninis tinklas gali būti netikslus ir gali tekti apmokyti tinklą su naujai sukurtu duomenų rinkiniu, kas gali užimti daug laiko ne vien apmokyti dirbtinį neuroninį tinklą, bet ir sukurti naują duomenų rinkinį, tad šiuo darbu siekiame ištirti kokioms transformacijomis yra labiausiai paveikiamas neuroninis tinklas ir bandysime optimizuoti jo testavimą.

8.1. Suformuluota užduotis

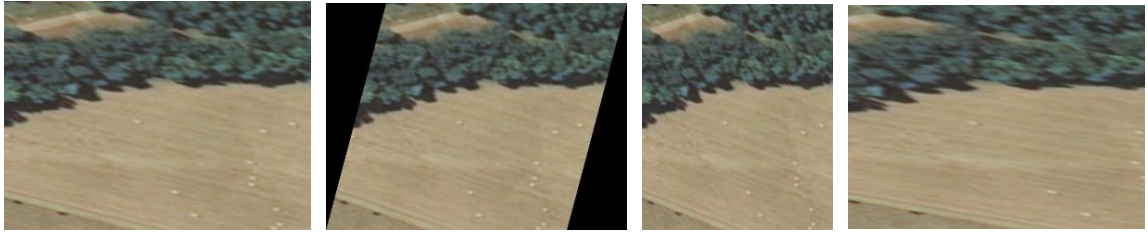
Praktinės dalies užduotis — atlikti analizę, kurią daro paveikslėlių transformacijos neuroninio tinklo modeliui.

8.2. Pasirinktos užduoties sprendimas

Išsikeltam uždaviniui spręsti siūloma pasitelkti analizės uždavinio sprendimo būdą. Tam įgyvendinti yra išsikelti uždaviniai kuriuos reikia įgyvendinti:

- Pritaikyti neuroninį tinklą prie savo naudojamų įrankių;
- Sukurti transformacijų generatorių, kuris pagal pasirinktus duomenis generuos reikiamas transformacijas su pateiktais parametrais ir apdoros paveikslėlius kuriuos naudosime neuroniniam tinklui analizuoti;
- Apmokyti neuroninio tinklo modelį skirtingomis epochomis;
- Apmokytą tinklą testuoti su įvairiomis transformacijomis ir analizuoti gautus rezultatus;

14 paveikslėlyje matome kaip generuojami paveikslėliai yra afininės transformacijos. Šis generavimo principas yra taikomas visoms transformacijomis, kurias naudojame. Kairiojoje pusėje matome ne transformuotą paveikslėlį. Dešiniau matomo kokią paveikslėlį gauname po transformacijos. Dar dešiniau matome paveikslėlį kuris yra apkarpytas su prarastais duomenimis, tai yra apkarpome taip jog nebeliktų paveikslėlyje juodų laukų. Tada mes jį išstampome taip jog atitiktų originalaus paveikslėlio matmenis.



14 pav. Pavaizdavimas kaip ruošiami duomenys tyrimui

Neuroninis tinklas yra apmokytas su adamo optimizavimo metodu ir naudota 3 duomenų rinkiniai (*angl. batch*). Neuroniniam tinklui apmokyti naudojame 1188 trejetų ir 105 trejetai naudojami validavimui. Visi paveikslėliai buvo 640 x 480 matmenų.

Tinklo testavimui buvo naudojama 100 trejetų kurie buvo naudojami visoms transformacijoms. Tyrime naudojamas tas pats apmokytas tinklas kiekvienai transformacijai. Tai reiškia jog jai neuroninis tinklas yra apmokytas penkiasdešimt epochų tai ir tiriama afininė transformacija, tai iš naujo nėra apmokomas neuroninis tinklas kitoms transformacijoms. Po atliktos transformacijos buvo paleistas neuroninis tinklas tikrinti paveikslėlių panašumus ir buvo gauti rezultatai kuriuos lyginant buvo tiriamas neuroninio tinklo atsparumas.

8.2.1. Tyrime naudojami įrankiai

Tyrime naudojamas modelis buvo parašytas naudojantis Keras 1.0.6 ir TensorFlow 2.0 neuroninio tinko bibliotekos. Su šiomis bibliotekomis galima greitai ir patogiai eksperimentuoti su neuroninio tinklo modeliais juos apmokyti ir testuoti. Programavimo kalba yra pasirinkta Pytjon 3.6.10.

Paveikslėlių transformacijoms generuoti yra naudojama OpenCV biblioteka, nes kaip ir Keras ir TensorFlow galima turi Python programavimo kalbos palaikymą, bei daugybę optimizuotų algoritmų.

Tyrimas buvo atliktas naudojantis techninę yra: GeForce RTX 2070 Max-Q vaizdo plokštę ir Intel I7 8750H – 6 branduolių procesorių ir 16 gb. dydžio operatyviąją atmintimi. Naudojantis Windows Home 64 bit. operacinę sistemą.

8.3. Tyrimas

8.3.1 Afininės transformacijos rezultatai

Afininei transformacijai naudoja metriką X kuris reiškia kiek procentų pasislinko paveikslėlis x ašyje tai yra pavaizduota 15 paveikslėlyje. Taigi jai X reikšmė yra lygi 10 procentų tai paveikslėlis pasitrauks 10 % savo pikselių į vieną pasirinktą pusę.

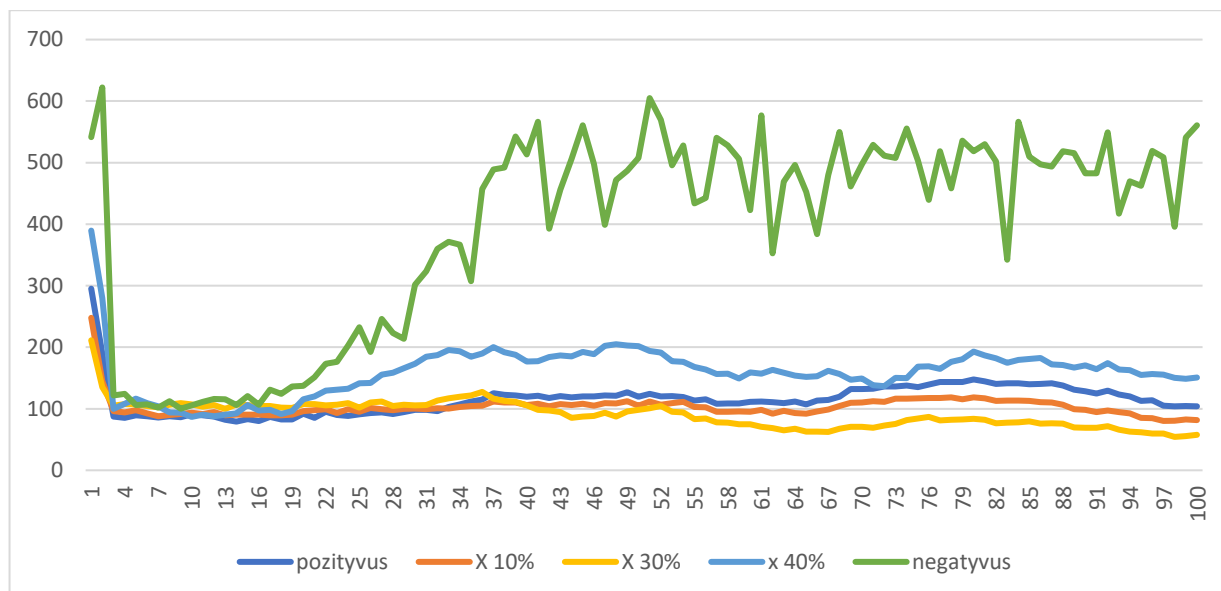


15 pav. Naudojama X metrika

1 lentelė. Neuroninio tinklo atsparumo afininei transformacijai rezultatai

epochos	pozityvus	X 10%	X 20%	X 30%	X 40%
5	94%	86%	69%	31%	11%
15	99%	94%	85%	83%	93%
25	87%	80%	65%	48%	27%
50	100%	100%	99%	95%	97%
75	100%	97%	92%	94%	97%
150	100%	96%	85	85%	90%

Pirmoje lentelėje matome neuroninio tinklo tikslumo rezultatus po gautus atlikus neuroninio tinklo atsparumo afininei transformacijai tyrimą. Galime pastebėti jog geriausius rezultatus rodė neuroninis tinklas apmokytas 50 epochų, o blogiausius neuroninis tinklas kuris yra apmokytas 5 epochomis. Pastebime jog 15 epochų apmokytas neuroninis tinklas yra atsparus lyginant jog jis nėra apmokytas su dideliu kiekiu epochų. Ypač jai lyginsime 15 epochų apmokytą tinklą su 25 epochų. 25 epochų apmokytas neuroninis tinklas atrodo jog yra persimokinęs ir neatsilaiko nuo afininės transformacijos.



16 pav. Paveikslėlių lyginimo rezultatai su neuroninio tinklo mokymų 50 epochų

Penkiasdešimt epochų apmokytas tinklas buvo atspariausias afininei transformacijai todėl pažvelgę į tinko rezultatų grafiką matome jog iš neuroninio tinklo gaunami panašumo parametrai po transformacijos yra nedaug nutolę nuo pradinių rezultatų. Tik kai afininės transformacijos parametras x pasiekė 40% reikšmę matomas žymus neuroninio tinklo gautų parametru padidėjimas.

8.3.2 Projekcinė transformacijos rezultatai

Projekcinės transformacijai naudoja metriką X kuris reiškia kiek procentų pasislinko paveikslėlis x ašyje tai pavaizduota 17 paveikslėlyje. Metrika yra naudojama panaši į afininės transformacijos metriką. Ją galima suprasti taip kiek procentų vienas šonas bus suspaustas.



17 pav. Naudojama X metrika

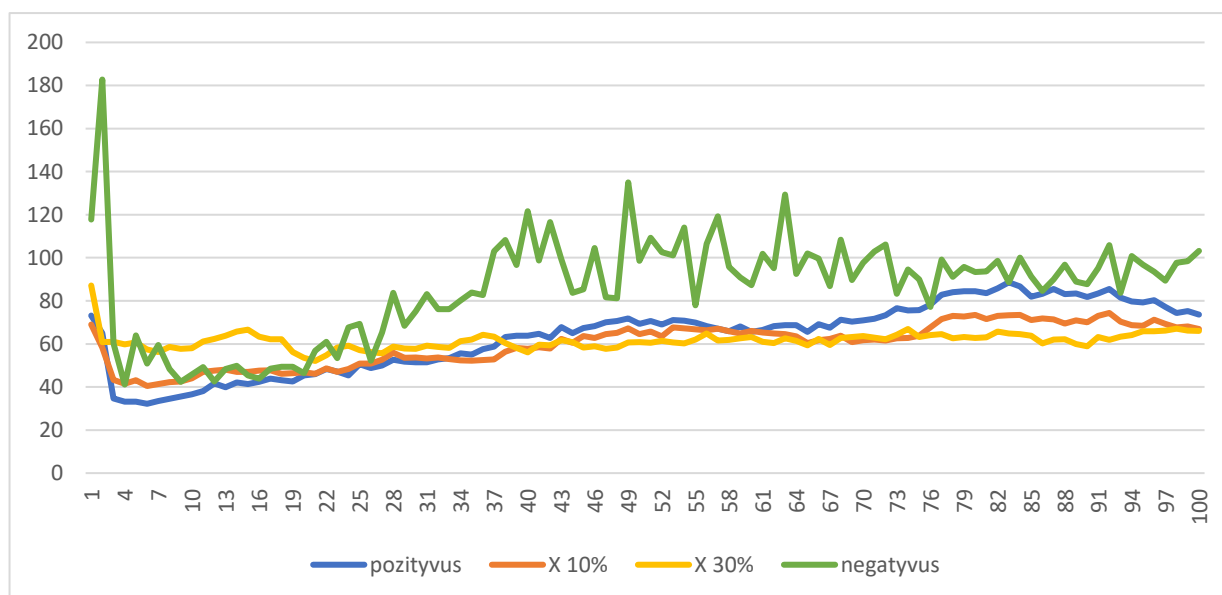
Antroje lentelėje yra pavaizduoti neuroninio tinklo atsparumo Projekcinei transformacijai rezultatai. Kaip ir afininės transformacijos tyrime galima matyti jog blogiausiai atsparus yra neuroninis tinklas apmokytas 5 epochomis. Po to seka tinklas kuris yra apmokytas 25 epochomis.

15, 50, 75, 150 aprodė panašius rezultatus, visais keturiais atvejais tinklas išsilaikė virš 80 procentų tikslumu.

2 lentelė. Neuroninio tinko atsparumo projekcinei transformacijai rezultatai

epochos	pozityvus	X 10%	X 20%	X 30%	X 40%
5	94%	94%	82%	65%	42%
15	99%	94%	87%	82%	93%
25	87%	87%	77%	58%	34%
50	100%	100%	88%	84%	84%
75	100%	93%	86%	84%	97%
150	100%	94%	85	84%	81%

Vienas atspariausių projekcinei transformacijai neuroninis tinklas buvo apmokytas 15 epochų jo išsamesni rezultatai yra pateikiami 18 paveikslėlyje. Šiame grafike matome jog trejetai kurie yra grafike rodomi nuo ketvirto iki trisdešimto numerio turėjo mažą parametro dydžio skirtumą skirtumą tarp teigiamo (panašaus) ir neigiamo (nepanašaus) paveikslėlių. Tai lėmė jog po transformacijos apmokytas neuroninis tinklas neatsilaikė ir suklydo nustatydamas paveikslėlių panašumą.



18 pav. Paveikslėlių lyginimo rezultatai su neuroninio tinklo mokymų 15 epochų

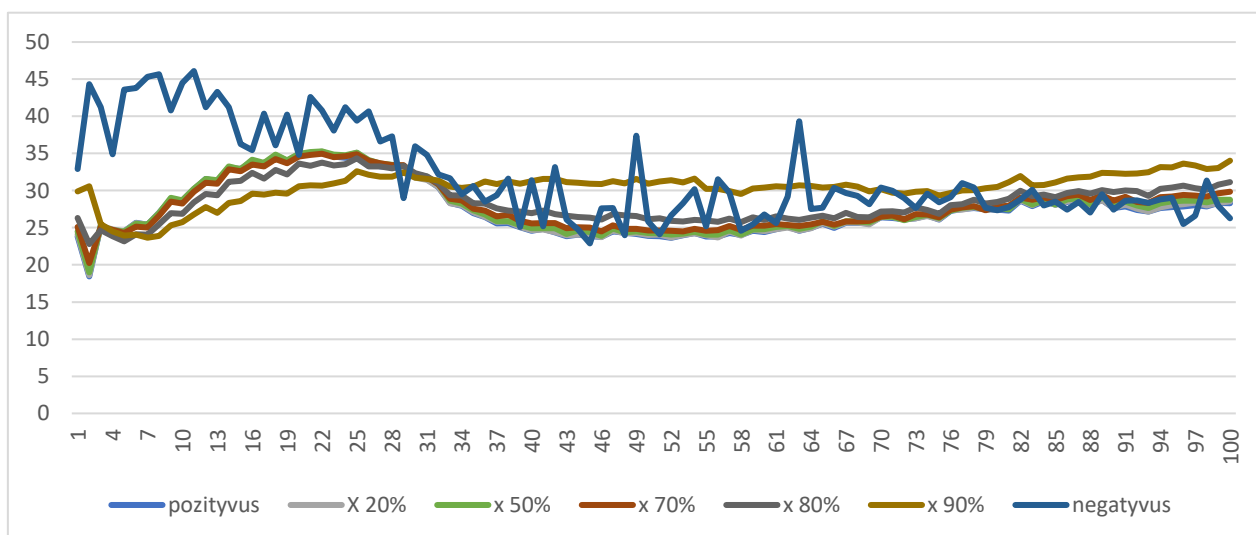
8.3.3 Suliejimo transformacijos rezultatai

Šioje suliejimo transformacijoje naudojamas parametras X kuris reiškia kiek procentaliai mažiname paveikslėlį ir taip prarandame informaciją. Paveikslėlį vėl išdidinus iki originalaus dydžio. Po šių veiksmų paveikslėlis gaunasi susiliejęs

3 lentelė. Neuroninio tinko atsparumo projekcinei transformacijai rezultatai

epochos	pozityvus	X 30%	X 40%	X 50%	X 60%	X 70%	X 80%
5	94%	92%	91%	90%	91%	89%	82%
15	99%	97%	97%	97%	97%	87%	82%
25	87%	83%	84%	83%	82%	76%	66%
50	100%	100%	100%	100%	100%	95%	87%
75	100%	98%	96%	93%	93%	91%	87%
150	100%	99%	99%	95%	95%	92%	84%

Trečioje lentelėje matome suliejimo transformacijos atlikto tyrimo rezultatus. Galime pastebėti jog su visomis pateiktomis epochomis apmokytas neuroninis tinklas atsilaukė transformacijai iki 60%. Atidžiau paanalizavus duomenis ir pridėjus jog paveikslėlio dydis yra 640x480 dydžio o neuroninio tinklo įeities sluoksnis yra 224x224. Tai atlikus skaičiavimus pastebime jog tik praradus 70% pikselių prarandame reikalingus pikselius kuriuos naudojame neuroninio tinklo modelio įeities sluoksnyje.

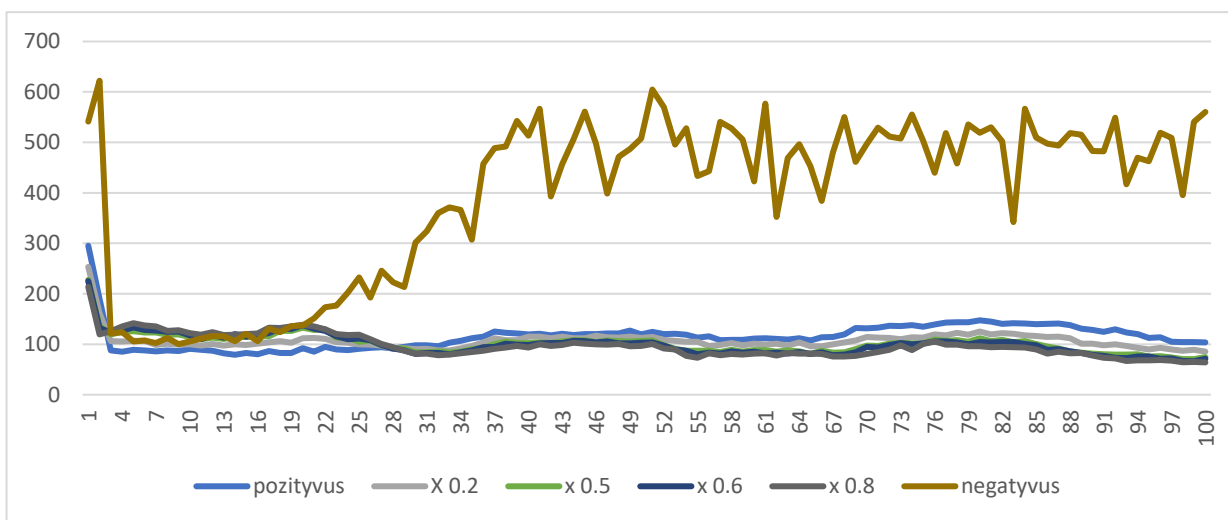


19 pav. Paveikslėlių lyginimo rezultatai su neuroninio tinklo mokymų 25 epochų

Devynioliktame paveikslėlyje matome išsamesnius 50 epochų apmokyto neuroninio tinklo atsparumo suliejimo transformacijai. Grafike matome jog tik kai X metrika pasiekia 80% tik tada neuroninio tinklo tikslumas pradeda mažėti. Su mažesniais parametrais neuroninis tinklas rezultatai po transformacijos nebuvo labai nutolę nuo pradinių reikšmių.

8.3.4 Statinės formos transformacijos rezultatai

Statinės formos transformacijai naudojame metrika X kuri yra formulės 17 X parametras. Vis didinant šį parametrą buvo gaunamos vis stipresnės transformacijos. Ir naudojantis apmokyto neuroninio tinklo modeliu buvo gaunami paveikslėlių panašumo rezultatai.



20 pav. Paveikslėlių lyginimo rezultatai su neuroninio tinklo mokymų 50 epochų

Dvidešimtame paveikslėlyje yra pavaizduotas 50 epochų apmokyto neuroninio tinklo išsamesni rezultatai. Kaip galime pastebėti su tokiu kiekiu epochų apmokytas neuroninis tinklas buvo visiškai atsparus statinės formos transformacijai, nes jo rezultatai gauti po transformacijos yra labai panašūs ir beveik nesiskiria nuo originalaus (pozityvaus) paveikslėlio.

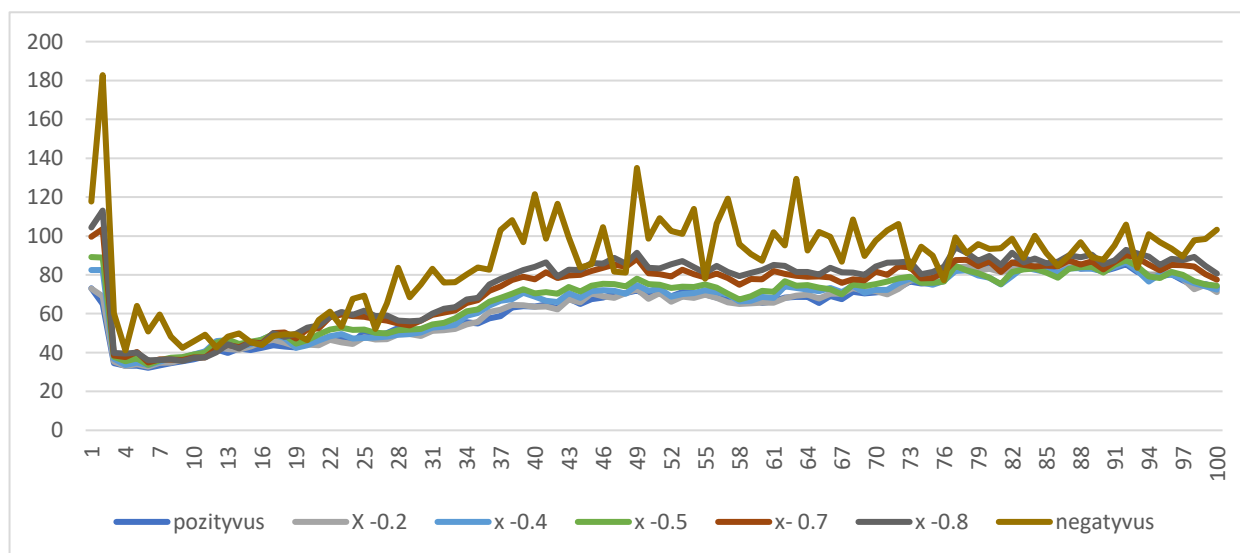
4 lentelė. Neuroninio tinklo atsparumo projekcinei transformacijai rezultatai

epochos	pozityvus	X 0.1	X 0.2	X 0.3	X 0.4	X 0.5	X 0.6
5	94%	94%	93%	86%	85%	82%	78%
15	99%	95%	88%	84%	83%	82%	82%
25	87%	81%	80%	79%	71%	69%	66%
50	100%	100%	99%	95%	92%	90%	87%
75	100%	93%	89%	87%	84%	83%	82%
150	100%	95%	92%	88%	86%	85%	82%

Ketvirtoje lentelėje atlikto neuroninio tinklo atsparumo statinės formos transformacijai tyrimo rezultatus. Šiai transformacijai neatspariausias apmokytas neuroninis tinklas buvo su 25 epochomis. Kaip ir prieš tai aprašytuose transformacijose geriausiai pasirodė su daugiausiai epochų apmokyti neuroninio tinklai.

8.3.5 Pagalvės formos transformacijos rezultatai

Pagalvės formos transformacijoje naudojame lygiai tokį pačią metriką X kaip ir naudojome priešingoje statinės formos transformacijoje, nes šiai transformacijai atvaizduoti naudojame tą pačią formulę numeriu 17.



21 pav. Paveikslėlių lyginimo rezultatai su neuroninio tinklo mokymų 15 epochų

Dvidešimt pirmame paveikslėlyje yra pateikti išsamesni rezultatai atliktos analizės neuroninio tinklo atsparumo pagalvės formos transformacijai. Kaip ir statinės formos

transformacijoje neuroninis tinklas ganėtinai yra atsparus nes rezultatai po transformacijos yra nežymiai pakitę nuo pozityvaus (netransformuoto) paveikslėlio rezultatų.

5 lentelė. Neuroninio tinko atsparumo projekcinei transformacijai rezultatai

epochos	pozityvus	X -0.1	X -0.2	X -0.3	X -0.4	X -0.5	X -0.6
5	94%	95%	94%	88%	90%	89%	87%
15	99%	98%	97%	97%	97%	94%	90%
25	87%	87%	86%	83%	80%	80%	74%
50	100%	100%	100%	100%	100%	100%	100%
75	100%	99%	100%	100%	98%	98%	100%
150	100%	100%	99%	96%	96%	95%	96%

Penktoje lentelėje galime matyti neuroninio tinklo atsparumo pagalvės formos transformacijai tyrimo rezultatus. Iš lentelės pateiktų duomenų galime pastebėti jog 15, 50, 75 ir 150 apmokytam neuroniniam tinklui transformacija įtakos neturėjo, tik 25 epochomis apmokytas neuroninis tinklas turėjo problemų atsilaikant šiai transformacijai.

Rezultatai ir išvados

Šio darbo metu buvo nagrinėjami dirbtiniai neuroninio tinklo medeliai ir atlikta analizė apmokyto dirbtinio neuroninio tinklo atsparumo transformacijoms. Išnagrinėtos paveikslėlių transformacijos. Šiam darbui įgyvendinti buvo atlikti šie darbai:

1. Buvo išnagrinėta literatūra apie dirbtinį neuroninį tinklą. Buvo išanalizuoti pagrindinės neuroninių tinklų dalys.
2. Buvo prisitaikytas neuroninio tinklo modelis ir jis apmokytas.
3. Sukurtas įrankis, kuris generuoja paveikslėlių transformacijas pagal pasirinktus parametrus ir sutvarko paveikslėlius taip jog juos galėtume naudoti testuojant neuroninio tinklo modeliui.
4. Atlikta apmokyto neuroninio tinklo atsparumo transformacijos gautų duomenų analizė.

Išvados:

1. Atsparumas transformacijoms buvo testuojamas ant įvairaus epochų skaičiaus apmokyto neuroninio tinklo modelio. Gavus rezultatus pastebėta jog geriausi rezultatai buvo gauti su penkiasdešimt ir daugiau epochų apmokyto neuroninio tinklo. Blogiausiai apsimokė tinklas su 25 epochų kiekiu, nes parodė blogiausius rezultatus. Tad galime sakyti jog jai norime jog neuroninis tinklas būtų atsparus paveikslėlių transformacijomis reikia tinklą apmokyti 50 ir daugiau epochų.
2. Atlikus analizę buvo padaryta išvada jog neuroninio tinklo modelis yra gan atsparus visoms tirtoms transformacijoms. Atspariausias buvo suliejimo transformacijai ir tai yra dėl to jog paveikslėlis tampa neryškus bet jo vaizdas nėra iškraipomas. Neatspariausias apmokytas neuroninis tinklas pagal gautus rezultatus buvo afininei, nes ji labiausiai iškraipo paveikslėlį iš likusių transformacijų.

Šaltiniai

- [HKC15] Samer Hijazi, Rishi Kumar, and Chris Rowen, IP Group, Cadence. Using Convolutional Neural Networks for Image Recognition. Cadence Design Systems, Inc. 2015 pp.2-4
- [WSL13] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, Ying Wu¹. Learning Fine-grained Image Similarity with Deep Ranking. Northwestern University, 2013.
- [LYZ17] Wentong Liao, Michael Ying Yang, Ni Zhan, Bodo Rosenhahn. Triplet-based Deep Similarity Learning for Person Re-Identification. Institute for Information Processing, 2017.
- [GV07] A. Gelžinis, A. Verikas. Neuroniniai tinklai ir neuroniniai skaičiavimai. Mokomoji knyga. Kauno technologijos universitetas, 2007
- [BK17] Andrius Berniukevičius, Eugenijus Kurilovas. Dirbtiniai neuroniniai tinklai personalizuotam mokymuisi. Lietuvos matematikos rinkinys, 2017.
- [Hs08] Haykin Simon. Library of Congress Cataloging-in-Publication Data. McMaster University, 2008.
- [NAEE18] Ahmed Nassar, Karim Amer, Reda ElHakim, Mohamed ElHelw. A Deep CNN-Based Framework For Enhanced Aerial Imagery Registration with Applications to UAV Geolocalization, 2018.
- [HBL17] Alexander Hermans, Lucas Beyer and Bastian Leibe. In Defense of the Triplet Loss for Person Re-Identification, 2017.
- [NSSKK18] Sergey Novoselov, Vadim Shchemelinin, Andrey Shulipa, Alexandr Kozlov and Ivan Kremnev. Triplet Loss Based Cosine Similarity Metric Learning for Text-independent Speaker Recognition, 2018.
- [OM18] Olivier Moindrot. Triplet Loss and Online Triplet Mining in TensorFlow, 2018.
- [MOA18] Marc-Olivier Arsenault,: Lossless triplet loss. a more efficient loss function for siamese nn (2018).
- [MCMMVB17] Zuheng Ming, Joseph Chazalon, Muhammad Muzzamil, Luqman Muriel Visani, Jean-Christophe Burie. Simple Triplet Loss Based on Intra/Inter-class Metric Learning for Face Verification, 2017.
- [VVVMRJ19] Vytautas Valaitis, Virginijus Marcinkevicius, Rokas Jurevicius. Learning Aerial Image Similarity using Triplet Networks 2019.

- [SSAF17] Sagar Sharma Activation Functions in Neural Networks 2017.
- [ANPLU18] Andrei Nicolae PLU: The Piecewise Linear Unit Activation Function 2018.
- [SZVDCN15] K. Simonyan, A. Zisserman Very Deep Convolutional Networks For Large-Scale Image 2015
- [RSCCA10] Richard Szeliski Computer Vision: Algorithms and Applications 2010.
- [AFSLM01] Andrew W. Fitzgibbon Simultaneous linear estimation of multiple view geometry and lens distortion 2001.