VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
STUDIJŲ PROGRAMA: INFORMATIKA

Dirbtinių neuroninių tinklų modelių paaiškinamumo ir intepretuojamumo tyrimai

Explainability and Interpretability of Artificial Neural Network Models

Baigiamasis magistro darbas

Atliko: Gabrielė Čiuladaitė
VU e-paštas: gabriele.ciuladaite@mif.stud.vu.lt

Vadovas: prof. dr. Olga Kurasova
Recenzentas: dr. Linas Litvinas

Vilnius – 2025

Santrauka

Apmokius dirbtinį neuroninį tinklą, dažnai tenka interpretuoti jį kaip juodąją dėžę, tačiau yra išskiriamos net kelios priežastys, kodėl yra svarbu mokėti paaiškinti ir interpretuoti dirbtinių neuroninių tinklų gaunamus rezultatus – gebant suprasti, kodėl modelis priėmė tam tikrą sprendimą, didėja sistemų patikimumas, atsiranda modelių tobulinimo galimybės bei naujas mokymosi šaltinis. LIME, SHAP ir Grad-CAM yra paaiškinamojo dirbtinio intelekto metodai, kurie gali padėti suteikti skaidrumo labai sudėtingiems ir neaiškiems modeliams, sužinoti modelių sprendimų priėmimo motyvus ir atpažinti priešiškas atakas, kurių metu paveikslėlis yra sugadinamas triukšmu, nematomu žmogaus akiai, tačiau paveikiančiu, kaip neuroninių tinklų modelis jį klasifikuoja. Pritaikius paaiškinamojo dirbtinio intelekto metodus (LIME, Grad-CAM, SHAP) galima vizualiai įvertinti požymius, lėmusius klasifikavimo rezultatus. Taip pat, pagal gautas SHAP skaitines pikselių svarbos vertes galima nustatyti tikimybę, jog paveikslėlis yra paveiktas triukšmu. Yra pastebima, jog triukšmu nepaveikto paveikslėlio požymiai yra reikšmingesni ir susitelkę į konkrečius paveikslėlyje matomus bruožus, o esant triukšmui sugeneruotas paaiškinimas mažiau koncentruojasi į keletą specifinių požymių, yra labiau atsitiktinai pasklidęs visame paveikslėlyje.

Raktiniai žodžiai: dirbtiniai neuroniniai tinklai, interpretuojamumas, paaiškinamumas, priešiškos atakos.

Summary

After training an artificial neural network, it is often necessary to interpret it as a black box. However, there are several reasons why it is important to be able to explain and interpret the results produced by artificial neural networks. Understanding why the model made a certain decision increases the reliability of systems, opens up possibilities for improving neural network models, and provides a new source of learning. LIME, SHAP, and Grad-CAM are explainable artificial intelligence methods that can help bring transparency to very complex and unclear models, uncover the reasoning behind the model's decisions, and detect adversarial attacks in which an image is corrupted with noise that is invisible to the human eye but affects how the neural network model classifies it. By applying explainable AI methods (LIME, Grad-CAM, SHAP), it is possible to visually evaluate the features that influenced the classification result. Additionally, based on the obtained SHAP numerical pixel importance values, it is possible to determine the likelihood that an image has been affected by a noise-based attack. It has been observed that features of a noise-free image are more significant and concentrated on specific visible elements in the image, whereas explanations generated for an attacked image are less focused on a few specific features, their importance is more randomly dispersed across the entire image.

Keywords: artificial neural networks, explainability, interpretability, adversarial attacks.

TURINYS

Įvadas

Šiais laikais vis dažniau vartojamos tokios sąvokos kaip dirbtinis intelektas, dirbtiniai neuroniniai tinklai ar mašininis mokymasis, siejamos su protu nesuvokiamomis naujovėmis, robotais, užvaldysiančiais pasaulį, baime būti pakeistais darbo rinkoje ar didžiausią perspektyvumą turinčia profesija. Nors ši sritis ir gąsdina kai kuriuos žmones, dirbtinio intelekto pagalba sukurti sprendimai suteikia naudos tiek kasdieniniame gyvenime, tiek darbo aplinkoje palengvindami įvairius procesus, susijusius su didelio duomenų kiekio apdorojimu, generavimu, sprendimų priėmimu, prognozavimu ir pan. Nuolat kuriami skaitmeniniai įrankiai ir išmanieji įrenginiai tampa vis labiau prieinami ir vis dažniau naudojami bei padeda greičiau dirbti su informacija, prisideda prie verslo valdymo ir procesų optimizavimo.

Viena iš labiausiai paplitusių dirbtinio intelekto sričių yra dirbtiniai neuroniniai tinklai (angl. *artificial neural networks*, ANN), savo struktūra ir veikimo principu imituojantys žmogaus smegenyse esančių neuronų vykdomas užduotis bandant apdoroti informaciją. Jie geba mokytis iš pateikiamų pavyzdžių ir šias žinias vėliau pritaikyti naujiems įvesties duomenims [PL16]. Egzistuoja gausybė jau apibrėžtų neuroninų tinklų architektūrų [HBA+14] – vienos jų labiau tinka apmokyti neuroninį tinklą dirbti su tekstiniais duomenimis, generuoti ar kitaip apdoroti sakinius ir pastraipas, kitos yra skirtos klasifikuoti arba segmentuoti paveikslėlius, suteikti jiems papildomų savybių, dar kitos – prognozuoti skaitines vertes pagal istorinius duomenis. Vykdant mokymo procesą, visiems neuroninių tinklų modeliams iteratyviai yra pateikiami įvesties duomenys ir vyksta įvairūs skaičiavimai, kurie lemia tai, jog pasibaigus mokymui, modelis geba generuoti naujus rezultatus pats. Apmokius dirbtinį neuroninį tinklą, dažnai tenka interpretuoti jį kaip juodąją dėžę (angl. *black box*). Kartais vizualiai galima įvertinti, ar gautas rezultatas atitinka lūkesčius bei paskaičiuoti reikšmę kitais būdais, tačiau tai apriboja automatizacijos procesus ir neleidžia pilnai pasitikėti dirbtinio intelekto modelio sugeneruotais rezultatais. Supratus kodėl ir kaip modelis priėmė tam tikrą sprendimą, būtų daug paprasčiau juo remtis ir įkomponuoti jo galimybes į įvairiose srityse naudojamas sistemas.

Vis plačiau naudojant dirbtinių neuroninių tinklų modelius, didėja ir atakų, vykdomų siekiant sukompromituoti modelių gaunamus rezultatus, skaičius. Vienos iš tokių atakų – priešiški duomenų rinkinio puolimai, kai rinkinyje esantys paveikslėliai yra sugadinami triukšmu taip, kad žmogus to pokyčio nepastebėtų, o neuroninio tinklo modelio klasifikavimo rezultatai išsikreiptų. Taip sugadinti paveikslėliai nebėra tinkami naudoti, o sprendimai, priimti atsižvelgiant į juos, tampa nepatikimi, todėl labai svarbu yra atskirti, kada paveikslėlis yra paveiktas triukšmo. Labiausiai tikėtina, jog tokios atakos bus vykdomos juodosios dėžės principu veikiančioje aplinkoje, kur galima matyti tik modelio rezultatus, todėl svarbu yra ieškoti būdų, kurie padėtų suprasti modelių sprendimų priėmimo motyvus. Paaiškinamasis dirbtinis intelektas yra dar visai neseniai išpopuliarėjusi tyrimų sritis, kurios tikslas yra naudojantis įvairiais įrankiais ir principais suteikti skaidrumo mašininio mokymosi modeliams. Šie įrankiai taip pat galėtų padėti atpažįstant priešiškus duomenų rinkinio puolimus (angl. *adversarial attacks*). Analizuojant paaiškinamojo dirbtinio intelekto metodų pagalba sugeneruotus rezultatus galima vizualiai įvertinti, ar svarbiausiais išskirti paveikslėlio požymiai sutampa su tais, kuriuos išskirtų žmogus, darantis sprendimą. Šią analizę galima praplės-

ti skaičiuojant pasirinktas metrikas triukšmu paveiktų ir nepaveiktų paveikslėlių pikselių svarbos vertėms ir pagal jas vertinti neuroninio tinklo veikimo patikimumą.

Darbo tikslas: Pritaikant paaiškinamojo dirbtinio intelekto metodus nustatyti, kaip galima atpažinti priešišką puolimą ir palyginti skirtingų metodų gaunamus paaiškinamumo ir interpretuojamumo rezultatus priešiškai užpultiems ir neužpultiems paveikslėliams.

Darbo uždaviniai:

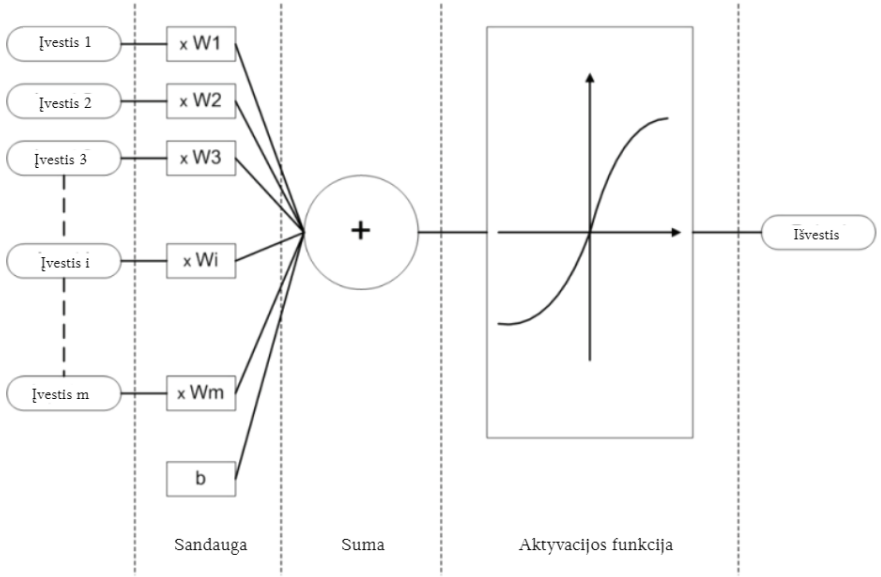
1. Naudojant paaaiškinamojo dirbtinio intelekto metodus išanalizuoti kaip iš anksto apmokytas dirbtinis neuroninis tinklas gauna rezultatus spręsdamas klasifikavimo užduotį paveikslėliams;
2. Palyginti rezultatus, gautus naudojant skirtingus paaaiškinamojo dirbtinio intelekto metodus;
3. Priešiškai užpulti kai kuriuos duomenų rinkinyje esančius paveikslėlius;
4. Naudojant paaaiškinamojo dirbtinio intelekto metodus išanalizuoti kaip iš anksto apmokytas dirbtinis neuroninis tinklas gauna rezultatus spręsdamas klasifikavimo užduotį priešiškai užpultiems paveikslėliams;
5. Palyginti rezultatus, gautus naudojant skirtingus paaaiškinamojo dirbtinio intelekto metodus priešiškai užpultiems ir neužpultiems paveikslėliams;
6. Nustatyti, kaip galima paaiškinamojo dirbtinio intelekto metodų pagalba atpažinti, ar paveikslėlis buvo priešiškai užpultas.

1. DIRBTINIAI NEURONINIAI TINKLAI

1.1. Dirbtinis neuronas

Dirbtinis neuronas, arba perceptronas, yra pagrindinė dirbtinio neuronino tinklo sudėtinė dalis, apibrėžiamas kaip paprastas trijų dalių matematinis modelis (funkcija) [KBK11]. Jis yra skirtas apdoroti gaunamus signalus x_m ir gauti reikšmę y. Warren McCulloch ir Walter Pitts 1943-aisiais metais pasiūlytas toks neurono modelis buvo sukurtas kaip realaus neurono žmogaus smegenyse imitacija siekiant atkartoti jo struktūrą, funkcionalumą ir sąveikos ypatumus efektyvesnėms informacijos apdorojimo sistemoms kurti [Gur97].

Kiekvienas neuronas turi x_m įvesčių (angl. *input*), vieną išvestį (angl. *output*) y bei slenksčio reikšmę b (angl. *bias*). Kiekviena įvestis neuroniniame tinkle yra dauginama iš kintamojo, vadinamu svoriu w_n. Vėliau šios sandaugos ir slenksčio reikšmė yra susumuojamos ir ši suma yra perduodama neurono aktyvacijos funkcijai, kurios rezultatas yra išvesties vertė y (1 pav.).



1 pav. Dirbtinio neurono veikimo principas [KBK11].

Matematiškai dirbtinį neuroną galima apibrėžti kaip [KBK11]:

$$y(k) = F(\sum_{i=0}^{m} W_i x_i + b), \tag{1}$$

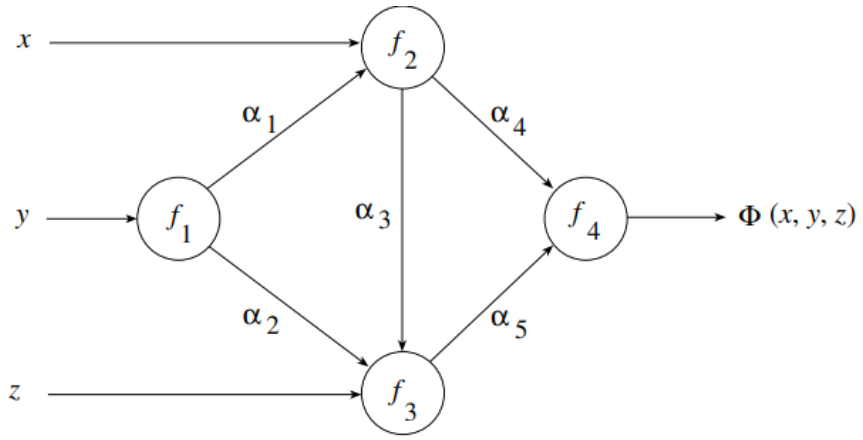
čia:

- x_i yra įvesties reikšmė, kur i yra nuo 0 iki m,
- W_i yra svorio reikšmė, kur i yra nuo 0 iki m,
- b yra slenkstis,
- F yra aktyvacijos funkcija,
- $y_i(k)$ yra išvesties reikšmė laike k.

1.2. Dirbtiniai neuroniniai tinklai

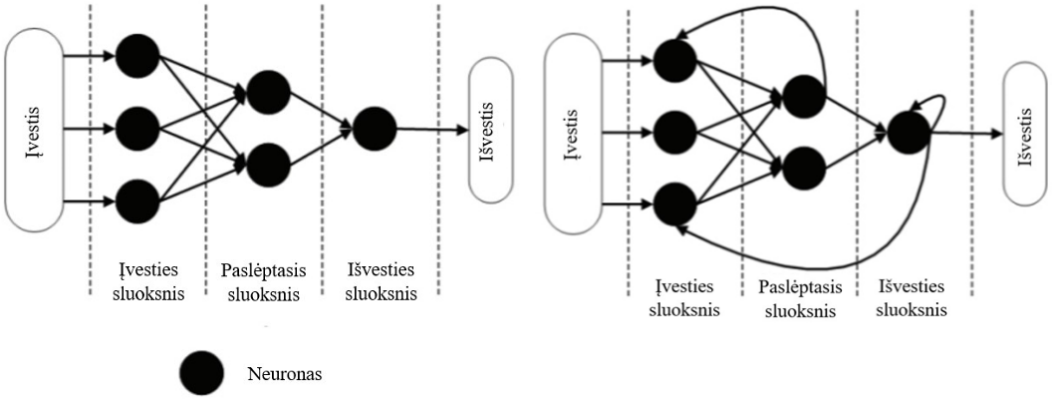
Sujungus du ar daugiau neuronų yra gaunamas dirbtinis neuroninis tinklas (DNT). DNT, priešingai nei vienas dirbtinis neuronas, yra pajėgus spręsti realaus gyvenimo uždavinius, reikalaujančius didelių resursų ir sudėtingų skaičiavimų, tokius kaip vaizdų atpažinimas ir klasifikavimas, natūralios kalbos apdorojimas, prekybos algoritmų kūrimas. DNT struktūrą sudaro trijų tipų sluoksniai: įvesties, paslėpti ir išvesties [KBK11]. Įvesties sluoksnis priima informaciją iš išorinių šaltinių ir siunčia ją gilyn į neuroninį tinklą. Paslėptas sluoksnis priima informaciją iš įvesties ar kitų paslėptų sluoksnių. Šiame sluoksnyje yra atliekami tinklo skaičiavimai. DNT gali turėti vieną ar daugiau paslėptų sluoksnių. Išvesties sluoksnis priima informaciją iš paslėptų tinklo sluoksnių ir perduoda ją išorei. Kuo daugiau paslėptų sluoksnių tinklas turi, tuo jis yra gilesnis ir pajėgesnis spręsti sudėtingesnes problemas, tačiau su kiekvienu papildomu sluoksniu didėja tinklo sudėtingumas ir veikimo trukmė, taip pat išauga persimokymo rizika [MW+14]. Neuroniniai tinklai, turintys daugiau nei vieną paslėptą sluoksnį, yra vadinami giliaisiais neuroniniais tinklais.

DNT sluoksniai susijungia į struktūrą pavaizduotą 2 paveikslėlyje. Šioje pavyzdinėje architektūroje neuroninis tinklas yra apibrėžiamas kaip ϕ, kuris yra įvertinamas su įvestimis (x, y, z). Kiekviename neurone yra implementuojamos primityvios aktyvacijos funkcijos (f_1, f_2, f_3, f_4), kurių rezultatai yra kombinuojami išvesčiai gauti. Keičiant tinklo svorius $(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5)$ gaunamos skirtingos išvestys. Taigi, neuroniniose tinkluose yra trys svarbūs kintamieji: neuronų struktūra (pasirinktos aktyvacijos funkcijos), neuronų išsidėstymas tinklo sluoksniuose (DNT topologija) ir optimalių svorių radimui naudojamas mokymosi algoritmas [Roj09].



2 pav. Dirbtinio neuroninio tinklo struktūros pavyzdys [Roj09].

Individualių dirbtinių neuronų tarpusavio sujungimo būdas DNT yra vadinamas topologija arba neuroninio tinklo architektūra. Kadangi neuronai gali būti sujungiami įvairiais būdais, egzistuoja daugybė galimų tinklo topologijų, kurios yra skirstomos į dvi pagrindines grupes: tiesioginio sklidimo (angl. *feedforward*) ir grįžtamojo ryšio (angl. *feedback*). 3 paveikslėlyje yra pavaizduotos šios dvi topologijos; kairėje pusėje pavaizduotas tiesioginio sklidimo neuroninis tinklas, kuriame informacija sklinda tik viena kryptimi, nuo įvesties iki išvesties, o dešinėje pusėje pavaizduotas grįžtamuosius ryšius turintinis neuroninis tinklas, kuriame dalis informacijos yra perduodama abiem kryptimis: nuo įvesties iki išvesties ir atgal [Abr05].



3 pav. Tiesioginio sklidimo ir grįžtamojo ryšio neuroninių tinklų topologijos [KBK11].

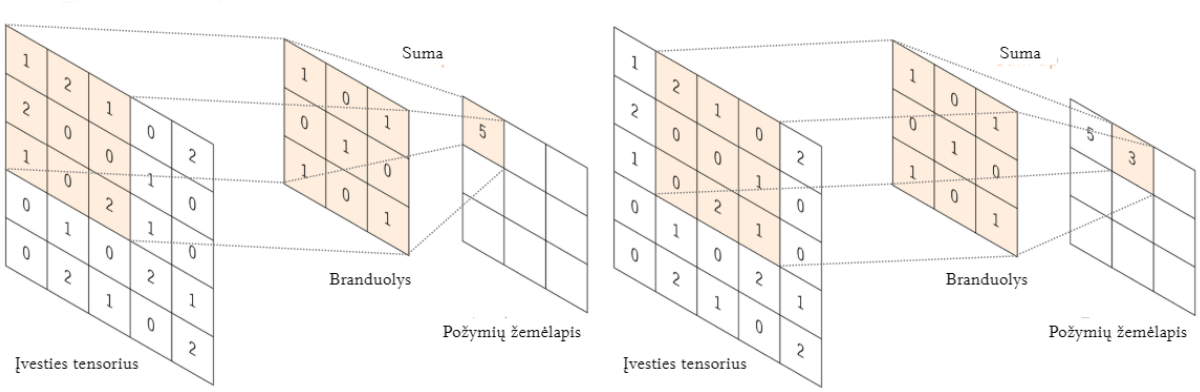
1.2.1. Konvoliuciniai neuroniniai tinklai

Konvoliuciniai neuroniniai tinklai (KNT) yra tiesinio sklidimo giliųjų neuroninių tinklų tipas, skirtas apdoroti duomenis, turinčius tinklelio (angl. *grid*) struktūrą, tokius kaip vaizdai. Šie tinklai yra sukurti automatiškai bei adaptyviai mokytis požymių erdvinių hierarchijų (angl. *spatial hierarchies*). KNT paprastai susideda iš trijų tipų sluoksnių (4 pav.): konvoliucijos (angl. *convolutional*), sujungimo (angl. *pooling*) ir pilnai sujungtų (angl. *fully connected*) sluoksnių. Pirmieji du, konvoliucijos ir sujungimo sluoksniai, atlieka požymių išgryninimo funkciją, o trečiasis, pilnai sujungtas sluoksnis, susieja gautus požymius su galutine išvestimi, tokia kaip klasifikacija [YND+18]. Tipinę KNT architektūrą sudaro pasikartojantys konvoliucijos ir sujungimo sluoksnių blokai ir vienas ar daugiau pilnai sujungtų sluoksnių. Kombinuojant konvoliucijos ir sujungimo operacijas iš pradžių išgaunami paprasti žemo lygio požymiai, pvz., linijos, o jas kartojant keletą kartų atpažįstami aukštesnio lygio požymiai, pvz., žmogaus veido bruožai.

4 pav. Paprasto KNT, skirto atpažinti ranka rašytus skaičius, architektūra, susidedanti iš penkių sluoksnių: įvesties sluoksnio, konvoliucijos sluoksnio, sujungimo sluoksnio ir dviejų pilnai sujungtų sluoksnių.

Konvoliucijos sluoksnis yra pagrindinis KNT architektūros komponentas, atliekantis požymių išgryninimą. Šis sluoksnis paprastai susideda iš tiesinių ir netiesinių operacijų derinio, tai yra, konvoliucijos operacijos ir aktyvacijos funkcijos. Konvoliucijos operacijos metu, branduoliai (angl. *kernel*) yra naudojami neuronams suorganizuoti į požymių žemėlapius (angl. *feature-maps*). Kitaip tariant, branduoliai yra filtrai, pagal kuriuos iš paveikslėlio yra išskiriami tam tikri požymiai – kiekvienas filtras aptinka tam tikrą atskirą požymį [ON15].

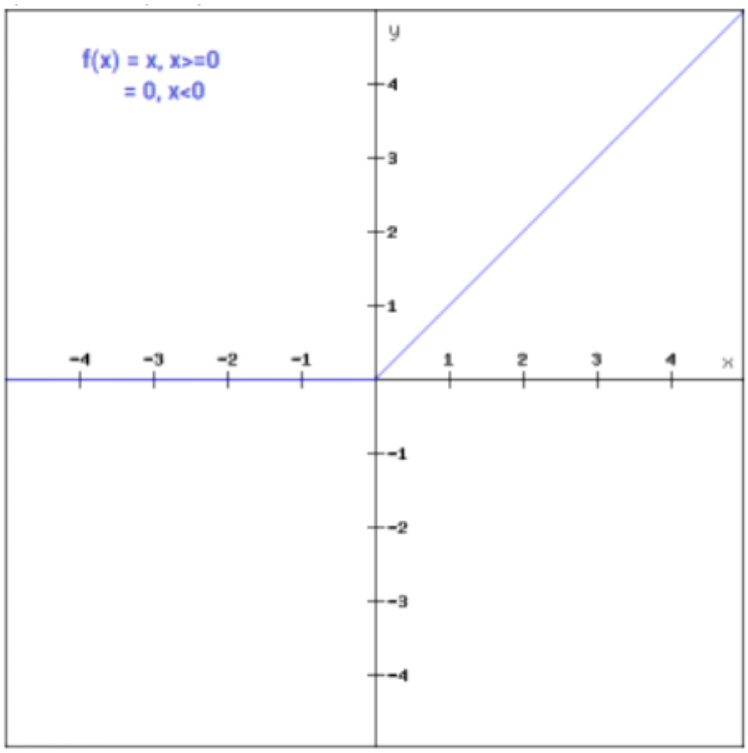
Įvestis į konvoliucijos sluoksnį dažniausiai yra dvimatis (2D) vaizdas, kurio pikselių reikšmės yra saugomos tinklelio pavidalu, vadinamu tensoriumi. Branduolys (taip pat vadinamas filtru) yra mažų parametrų tinklelis. Šis branduolys yra optimizuojamas mokymo proceso metu. Dažnas branduolio dydis yra 3x3, tačiau kartais gali būti pasirenkamas ir 5×5 ar 7×7 matmenų branduolys [YND+18]. Konvoliucijos operacijos metu branduolys taikomas kiekvienai įvesties vaizdo vietai (5 pav.), atliekant elementų sandaugą tarp branduolio ir atitinkamos įvesties tinklelio dalies. Sandaugos rezultatai sumuojami ir taip gaunama reikšmė, kuri yra įrašoma į atitinkamą poziciją išvesties požymių žemėlapyje. Norint išlaikyti įvesties tinklelio matmenis, dažnai naudojamas nulinis pildymas (angl. *zero padding*). Tai reiškia, kad įvesties tinklelio kraštai apjuosiami nulių eilutėmis ir stulpeliais prieš atliekant konvoliucijos operaciją [LLY+21].



5 pav. Konvoliucijos operacijos pavyzdys, kur branduolio dydis yra 3x3, žingsnis yra 1 ir nėra jokio pildymo [YND+18].

Atstumas tarp dviejų gretimų branduolio pozicijų vadinamas žingsniu (angl. *step*). Žingsnis nustato per kiek pikselių branduolys judės įvesties matricoje. Dažniausiai pasirenkamas žingsnio dydis yra 1, tačiau didesnių matmenų vaizdams galima naudoti didesnį žingsnį, nes tokiuose vaizduose požymiai irgi bus didesni. Didesnio žingsnio pasirinkimas leidžia pagreitinti tinklo skaičiavimus neprarandant požymių. Tačiau, jei bus pasirinktas per didelis žingsnis apdorojamam paveikslėliui, atsiras didesnė tikimybė praleisti svarbius požymius. Branduolių dydis, branduolių skaičius, pildymas ir žingsnis yra KNT hiperparametrai, kuriuos reikia nustatyti prieš pradedant mokymo procesą. Po konvoliucijos operacijos yra gaunamas požymių žemėlapis, kuris parodo, kuriose vietose įvesties vaizde buvo aptikti tam tikri požymiai. Dažniausiai po konvoliucijos operacijos taikoma netiesinė aktyvacijos funkcija, pavyzdžiui, ReLU (angl. *Rectified Linear Unit*) – taip tinklas prisitaiko atpažinti sudėtingesnius šablonus (angl. *patterns*) [GWK+18]. ReLU funkcija (6 pav.) yra dažniausiai naudojama aktyvacijos funkcija giliuosiuose neuroniniuose tinkluose dėl savo paprastumo ir efektyvumo. Ši funkcija grąžina 0 neigiamoms įvesties vertėms ir įvesties reikšmę teigiamoms vertėms, taigi, šios funkcijos išvestys yra intervale $[0, \infty)$. Matematiškai ši funkcija yra apibrėžiama, kaip [DSC22]:

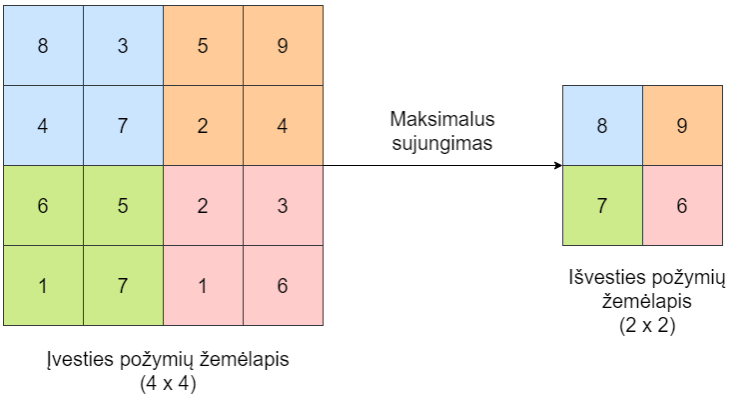
$$ReLU(x) = max(0,x) = \begin{cases} x & \text{jei } x \geq 0 \\ 0 & \text{jei } x < 0 \end{cases} \tag{2}$$



6 pav. ReLU aktyvacijos funkcijos grafikas [SSA17].

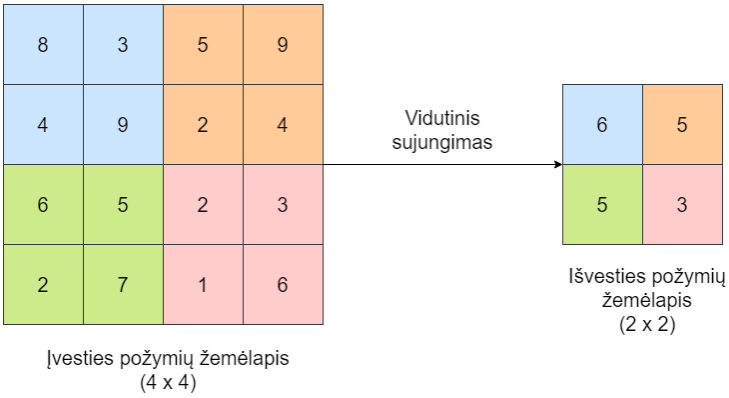
Sujungimo sluoksniuose vyksta paveikslėlio erdvinių matmenų (aukščio ir pločio) apimties mažinimas. Šis sumažinimas įveda toleranciją mažiems poslinkiams ir iškraipymams įvesties duomenyse. Tai reiškia, kad maži požymių padėties pokyčiai (pvz. daikto pasukimas į šoną) neturės reikšmingos įtakos KNT rezultatams. Tai svarbu tokioms užduotims kaip vaizdų atpažinimas, kur tas pats objektas paveikslėliuose gali būti skirtingose padėtyse. Sumažinant požymių žemėlapių erdvinius matmenis, taip pat sumažinamas išmokstamų parametrų skaičius tinkle [YXF+18]. Tai palengvina skaičiavimus ir padeda išvengti KNT persimokymo, nes modelis tampa paprastesnis ir mažiau linkęs įsiminti mokymo duomenis [LLY+21]. Sujungimo sluoksniuose nėra jokių išmokstamų parametrų. Skirtingai nuo konvoliucijos sluoksnių, kurie mokymo metu išmoksta filtrų svorius, sumažinimo sluoksniai atlieka fiksuotas operacijas pagal iš anksto nustatytus hiperparametrus. Dažniausiai naudojamos sujungimo operacijų rūšys yra maksimalus sujungimas (angl. *max pooling*) ir vidutinis sujungimas (angl. *average pooling*).

Maksimalaus sujungimo metu, pasirinkto dydžio branduolys slenka per įvesties požymių žemėlapį per nustatytą žingsnį ir kiekvieną kartą taikant šį filtrą yra išrenkama didžiausia vertė ir įrašoma į išvesties požymių žemėlapį (7 pav.), taip išsaugojant svarbiausias savybes. Daugumoje KNT sujungimo sluoksniai yra maksimalaus sujungimo sluoksniai su 2×2 dydžio branduoliais su žingsnio verte 2. Tai sumažina požymių žemėlapį iki 25 % pradinio dydžio. Taip pat gali būti naudojamas persidengiantis sujungimas (angl. *overlapping pooling*), kai žingsnis nustatomas 2, o branduolio dydis – 3. Dėl sujungimo sluoksnio destruktyvumo, branduolio dydis virš 3 dažniausiai labai sumažina modelio našumą [ON15].



7 pav. Maksimalaus sujungimo pavyzdys, kur branduolio dydis yra 2×2, žingsnis yra 2.

Vidutinis sujungimas (8 pav.) vyksta panašiu principu į maksimalų sujungimą, tačiau vietoj to, kad būtų išrenkama didžiausia vertė filtre, yra suskaičiuojamas visų verčių vidurkis [BSC21].



8 pav. Vidutinio sujungimo pavyzdys, kur branduolio dydis yra 2×2, žingsnis yra 2.

Paskutinio konvoliucijos ar sujungimo sluoksnio išvesties požymių žemėlapiai paprastai yra išlyginami, t. y. transformuojami į vienmatį skaičių masyvą (arba vektorių), ir perduodami vienam ar daugiau pilnai sujungtų sluoksnių, kuriuose kiekviena įvestis yra sujungta su kiekviena išvestimi per išmokstamą svorį. Kai konvoliucijos sluoksniai išgauna požymius ir sujungimo sluoksniai sumažina juos, jie yra susiejami su galutinėmis tinklo išvestimis, tokiomis kaip tikimybės kiekvienai klasei klasifikavimo užduotyse [YND+18]. Po kiekvieno pilnai sujungto sluoksnio yra naudojama netiesinė aktyvacijos funkcija, tokia kaip ReLU. Paskutiniame pilnai sujungtame sluoksnyje taikoma aktyvacijos funkcija paprastai skiriasi nuo kitų. Tinkama aktyvacijos funkcija turi būti pasirinkta atsižvelgiant į sprendžiamą uždavinį.

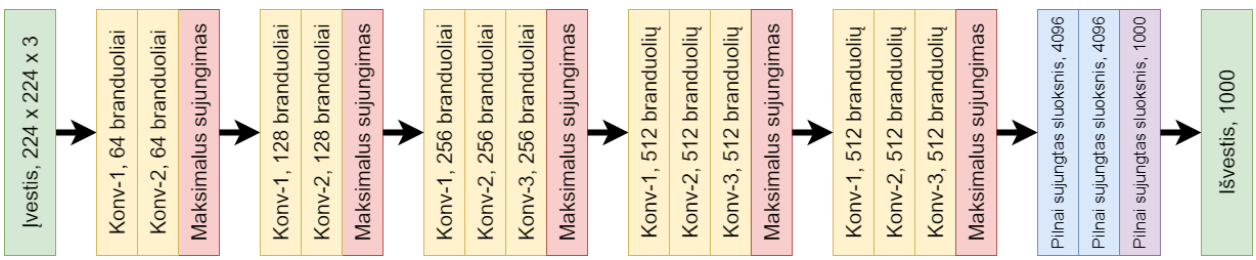
1.2.2. VGG16

VGG16 modelis [SZ15] yra konvoliucinio neuroninio tinklo architektūra, sukurta Oksfordo universiteto vaizdinės geometrijos grupės (angl. *Visual Geometry Group*) 2014 metais *ImageNet* didelio masto vaizdų atpažinimo konkursui (angl. *ImageNet Large Scale Visual Recognition Challenge*, ILSVRC) [Ima14]. ILSVRC yra kasmetinis kompiuterinės regos konkursas [RDS+15], kuriame komandos sprendžia įvairius uždavinius, įskaitant vaizdų klasifikavimą ir objektų aptikimą

paveikslėliuose. VGG16 šiame konkurse pasiekė aukštus rezultatus abiejose užduotyse, klasifikuodamas vaizdus į 1000 kategorijų ir aptikdamas objektus iš 200 klasių.

VGG16 yra paprastas ir efektyvus modelis, gebantis pasiekti gerus rezultatus įvairiose kompiuterinės regos užduotyse, įskaitant vaizdų klasifikavimą [KP20] [AA22] ir objektų atpažinimą [AAS+22]. Jis išsiskiria savo gilumu, susidedančiu iš 21 sluoksnio: 13 konvoliucinių sluoksnių, 5 sujungimo sluoksnių ir 3 pilnai sujungtų sluoksnių, tačiau, tik 16 sluoksnių turi išmokstamus parametrus (konvoliucijos ir pilnai sujungti sluoksniai), dėl to modelis ir vadinamas VGG16. Modelio architektūrą sudaro konvoliucinių sluoksnių blokai, tarp kurių yra maksimalaus sujungimo sluoksniai su palaipsniui didėjančiu gilumu. Šis dizainas leidžia modeliui išmokti sudėtingus vaizdų požymius, užtikrinant tikslias prognozes.

Detaliau, VGG16 architektūra [SZ15] susideda iš įvesties sluoksnio, 5 konvoliucinių sluoksnių ir maksimalaus sujungimo sluoksnių blokų, 3 pilnai sujungtų sluoksnių ir išvesties sluoksnio. VGG16 įvestis yra 224x224 dydžio tensorius su 3 RGB kanalais. Pirmasis paslėptų sluoksnių blokas yra sudarytas iš dviejų iš eilės einančių konvoliucinų sluoksnių, kurių kiekvienas turi 64 3×3 dydžio branduolius, ir vieno maksimalaus sujungimo sluoksnio su 2×2 dydžio branduoliais ir žingsniu 2. Antrasis paslėptų sluoksnių blokas yra sudarytas iš dviejų iš eilės einančių konvoliucinų sluoksnių, kurių kiekvienas turi 128 3×3 dydžio branduolius, ir vieno maksimalaus sujungimo sluoksnio su 2×2 dydžio branduoliais ir žingsniu 2. Trečiasis blokas yra sudarytas iš trijų iš eilės einančių konvoliucinų sluoksnių, kurių kiekvienas turi 256 3×3 dydžio branduolius, ir vieno maksimalaus sujungimo sluoksnio su 2×2 dydžio branduoliais ir žingsniu 2. Ketvirtasis ir penktasis paslėptų sluoksnių blokai yra abu sudaryti iš trijų iš eilės einančių konvoliucinų sluoksnių, kurių kiekvienas turi 512 3×3 dydžio branduolius, ir vieno maksimalaus sujungimo sluoksnio su 2×2 dydžio branduoliais ir žingsniu 2. Visuose konvoliucijos sluoksniuose yra naudojamas branduolio žingsnis 1 ir ReLU aktyvacijos funkcija. Pasirinkimas naudoti tris sluoksnius su 3×3 dydžio branduoliais vietoj vieno sluoksnio su 7×7 dydžio branduoliais VGG16 modelyje žymiai sumažina apmokomų parametrų skaičių ir leidžia modeliui išmokti atpažinti sudėtingesnius požymius. Po šių sluoksnių blokų seka 3 pilnai sujungti sluoksniai. Pirmojo pilnai sujungto sluoksnio įvesties dydis yra 25088, o išvesties dydis yra 4096, jame naudojama ReLU aktyvacijos funkcija. Antrasis pilnai sujungtas sluoksnis turi tiek pat neuronų, kiek ir pirmasis, dėl to išvesties dydis nekinta. Aktyvacijos funkcija šiame sluoksnyje taip pat ReLU. Trečiojo pilnai sujungto sluoksnio išvesties dydis yra 1000, tiek neuronų, į kiek klasių klasifikuota ILSVRC konkurse. Šis sluoksnis turi *Softmax* aktyvacijos funkciją. Iš viso, VGG16 tinklas turi 138 milijonus mokomų parametrų. VGG16 architektūra pavaizduota 9 paveikslėlyje.



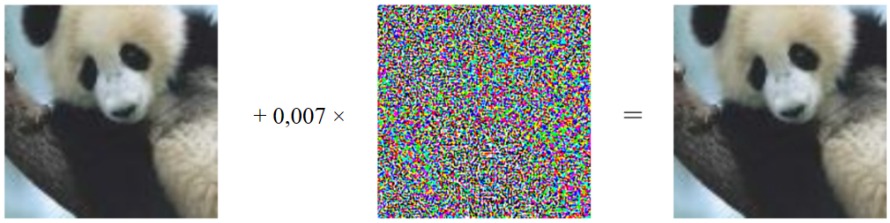
9 pav. VGG16 tinklo architektūra [SZ15].

Iš anksto apmokyto (angl. *pre-trained*) VGG16 modelio svoriai yra laisvai prieinami. VGG16 buvo mokomas su *ImageNet* duomenų rinkiniu. *ImageNet* yra viešai prieinamas duomenų rinkinys, kuris yra nuolat pildomas naujais vaizdais – jame tikimasi ateityje pasiekti 50 milijonų švariai sužymėtų pilnos rezoliucijos vaizdų [DDS+09]. VGG16 mokymui naudoti 1,2 milijonų paveikslėlių [SZ15], priklausančių įvairioms kategorijoms, tokioms kaip įvairių rūšių gyvūnai, transporto priemonės. Naudojant iš anksto apmokytą neuroninį tinklą sutaupomi dideli resursų kiekiai, kurie kitaip būtų naudojami tinklui apmokyti. Modelis, apmokytas su dideliu kiekiu duomenų, taip pat dažnai gaus geresnius rezultatus, nes jis sugebės atpažinti daugybę įvairių sudėtingesnių požymių. Iš anksto išmokyti modeliai gali būti tikslinami ir lengvai keičiami pagal užduoties specifiką, taip pasiekiant tikslesnius ir patikimesnius rezultatus per trumpesnį laiką [ZQD+20].

2. PRIEŠIŠKI PUOLIMAI

Vis plačiau naudojant dirbtinių neuroninių tinklų modelius, didėja ir atakų, vykdomų siekiant sukompromituoti modelių gaunamus rezultatus, skaičius. Vienas iš būdų, kaip galima įvykdyti tokį puolimą yra sugadinti duomenis, kuriuos DNT naudoja kaip įvestį rezultatams gauti. Priešiški puolimai yra skirstomi į baltosios dėžės ir juodosios dėžės principu veikiančius puolimus [HGJ23]. Baltosios dėžės puolimai vykdomi tada, kai sistemos architektūra ir parametrai yra žinomi, todėl tokios atakos yra mažiau tikėtinos. Prieiga prie parametrų leidžia pasiekti nuostolių funkcijos gradientą, kurį lengva panaudoti trikdžių generavimui. Tuo tarpu vykdant juodosios dėžės puolimą, yra pasiekiamos tik tinklo išvestys, tačiau nėra žinoma vidinė modelio struktūra. Priešiški puolimai, įvykdyti duomenų rinkiniui, dažniausiai yra skirti paveikslėlius klasifikuojantiems neuroniniams tinklams, kurių metu paveikslėlis yra sugadinamas taip, jog žmogus to pokyčio nepastebi, o neuroninių tinklų modelio klasifikavimo rezultatai išsikreipia [HGJ23]. Puolimai taip pat gali būti tiksliniai (angl. *targeted*) ir netiksliniai (angl. *non-targeted*). Vykdant netikslinius puolimus, duomenys yra sugadinami taip, jog modelis pradėtų juos klasifikuoti į bet kurią kitą klasę, tačiau vykdant tikslinius puolimus yra bandoma suklaidinti modelį suklasifikuoti duomenis į tam tikrą iš anksto užpuolikui žinomą specifinę klasę [HGJ23].

Priešišką puolimą galima įvykdyti pasitelkiant triukšmą. Tokio puolimo pavyzdys yra pateiktas 10 paveikslėlyje. *GoogLeNet* iš pradžių atpažino, jog paveikslėlyje yra pavaizduota panda, tačiau į paveikslėlį pridėjus triukšmo, modelio spėjimo rezultatai pasikeitė ir net su 99,3 % tikslumu jis spėjo, jog tame paveikslėlyje yra pavaizduotas gibonas [GSS15].



10 pav. *GoogLeNet* neurininio tinklo klasifikavimo rezultatai pagal [GSS15].

Vienas iš būdų, pasitelkiančių triukšmą, yra greito gradiento ženklo metodas (angl. *fast gradient sign method*, FGSM) [SD23]. FGSM yra baltosios dėžės principu veikianti ataka – tai reiškia, kad modelio architektūra ir parametrai turi būti žinomi. Jis sutrikdo įvesties duomenis pridedant nedidelį triukšmo kiekį, atsižvelgiant į nuostolių funcijos gradientą (angl. *gradient of the loss*). Šis metodas yra paprastas ir efektyvus bei reikalauja minimalių skaičiavimo resursų [GSS15]. Jo išvesties rezultatas yra vaizdas, kuris žmogaus akimis atrodo kaip originalus paveikslėlis, tačiau neuroninio tinklo klasifikuojamas klaidingai. FGSM gali būti aprašomas funkcija:

$$adv_x = x + \varepsilon sign(\nabla_x J(\theta, x, y)), \tag{3}$$

čia adv_x yra triukšmu paveiktas paveikslėlis, x – pradinis paveikslėlis, y – originalaus paveikslėlio tikroji klasė, ε – triukšmo intensyvumas, išreikštas maža trupmenine verte, iš kurios yra dauginami gradientai, jog būtų sukurti trikdžiai (šios vertės turi būti pakankamai mažos, kad būtų nepastebimos žmogui), θ – neuroninio tinklo modelis, naudojamas klasifikacijai ir J – nuostolių funkcija [SD23].

FGSM metodo veikimą galima būtų apibrėžti šiais žingsniais:

1. Pasirenkamas paveikslėlis pateikiamas neuroniniam tinklui, kuris pateikia spėjimus.
2. Apskaičiuojama nuostolių funkcijos reikšmė pagal žinomą tikrąją klasę.
3. Apskaičiuojamas nuostolių funkcijos gradientas.
4. Sugeneruojami trikdžiai, skaičiuojant ženklo (angl. *sign*) funkcijos reikšmę nuostolių funkcijos gradientui ir dauginant ją iš konstantos, kontroliuojančios trikdymo dydį.
5. Trikdžiai yra pritaikomi įvesties paveikslėliui.

Dar vienas priešiško puolimo metodas yra projekcinio gradientinio nusileidimo (angl. *Projected Gradient Descent*, PGD) ataka. Ši ataka yra patobulinta iteratyvi greito gradiento ženklo metodo (FGSM) versija ir laikoma viena stipriausių pirmojo lygio (angl. *first-order*) atakų [MMS+19].

PGD ataka gali būti aprašoma funkcija:

$$x_{t+1} = \Pi_{x+\mathcal{S}} \left(x_t + \alpha \cdot \text{sign} \left(\nabla_x L(\theta, x_t, y) \right) \right), \tag{4}$$

čia x_{t+1} yra sugeneruotas priešiškas pavyzdys t-ame žingsnyje, α – žingsnio dydis, $\nabla_x L(\theta, x_t, y)$ – nuostolio funkcijos L gradientas pagal įvestį x_t esant modelio parametrams θ ir tikrajai klasei y, $\text{sign}(\cdot)$ – ženklo funkcija (grąžina -1 arba $+1$ kiekvienam komponentui, taip nustatant atakos kryptį), $\Pi_{x+\mathcal{S}}(\cdot)$ – projekcijos operatorius, kuris užtikrina, kad trikdymas liktų leidžiamoje srityje aplink originalų įvesties pavyzdį x [MMS+19].

PGD metodo veikimo principą galima apibrėžti šiais žingsniais:

1. Pasirenkamas įvesties paveikslėlis.
2. Pasirenkamas iteracijų skaičius ir pradedamas iteracijų ciklas:
   (a) Apskaičiuojamas nuostolio gradientas pagal pasirinktą įvestį naudojant nuostolio funkciją. Šis gradientas parodo, kiek modelio nuostolis padidės, jei įvestis pasikeis, ir į kurią pusę ją reikėtų pakoreguoti.
   (b) Naudojant mažą žingsnio dydį yra atnaujinama įvestis (uždedamas triukšmas) nustačius gradientų kryptį, kuri didina nuostolį.
   (c) Užtikrinama, kad perturbuota įvestis liktų leistinose iškraipymo ribose. Tai daroma siekiant įsitikinti, jog perturbuota įvestis neperžengia maksimalių leidžiamų pokyčių nuo originalios įvesties.
3. Po nurodyto iteracijų skaičiaus, grąžinamas galutinis priešiškas pavyzdys.

PGD ataka pasižymi atsparių priešiškų pavyzdžių generacija ir mažesniu jautrumu hiperparametrų pasirinkimui, tačiau reikalauja didesnių skaičiavimo sąnaudų nei vieno žingsnio metodai (FGSM) dėl pasitelkiamų kelių gradientinio nusileidimo skaičiavimo iteracijų.

Taikant trikdžius yra siekiama, jog nuostolių funkcijos reikšmė ženkliai padidėtų ir paveikslėlis būtų klasifikuojamas klaidingai, tačiau pasikeitimas liktų nepastebimas. Konstantos, kontro-

liuojančios trikdymo dydį, nustatymas yra eksperimentinis uždavinys. Didesnė jos reikšmė lemia didesnį trikdymą, tačiau triukšmo pritaikymas skirtingiems paveikslėliams sugeneruoja skirtingus rezultatus, todėl jos reikšmę reikėtų pritaikyti naudojamam duomenų rinkiniui.

Triukšmu sugadinti paveikslėliai nebėra tinkami naudoti, o sprendimai, priimti atsižvelgiant į juos, tampa nepatikimi, todėl labai svarbu yra atskirti, kada paveisklėlis yra paveiktas triukšmu. Tradiciniai metodai, skirti mašininio mokymosi modeliams padaryti juos atsparesniais (angl. *robust*), pvz., svorio mažinimas (angl. *weight decay*) ir išmetimo sluoksnių (angl. *dropout*) pridėjimas, paprastai nesuteikia praktinės apsaugos nuo priešiškų puolimų, tačiau egzistuoja metodų, kuriais yra bandoma apsiginti nuo jų:

- Priešiškas mokymas (angl. *adversarial training*) – yra sugeneruojamas didelis kiekis priešiškai užpultų paveikslėlių ir modelis yra mokomas su užpultais ir neužpultais paveikslėliais kartu [BLZ+21]. Šio metodo paprastai nepakanka visoms atakoms sustabdyti, nes galimų atakų diapazonas yra per platus ir jų negalima sugeneruoti iš anksto.
- Gynybinė distiliacija (angl. *defensive distillation*) – tai strategija, apimanti dviejų modelių apmokymą, kai antrajam modeliui yra perduodamos pirmojo modelio žinios. Pirmasis neuroninis tinklas yra apmokomas išvesti tikimybes skirtingoms klasėms, o ne griežtai nuspręsti kurią klasę priskirti duomenims. Šios tikimybės yra vėliau naudojamos antram modeliui mokyti, kurio tikslas yra imituoti originalaus modelio elgesį, tačiau ne tiesiogiai naudojant originalius mokymo duomenis, taip mažinant jautrių duomenų atskleidimo riziką. Ši technika yra ypatingai vertinga scenarijams, kai privatumas ir saugumas yra svarbūs [PMW+16].

Labiausiai tikėtina, jog priešiškos atakos bus vykdomos juodosios dėžės principu veikiančioje aplinkoje, kur galima matyti tik modelio rezultatus [HGJ23], todėl paaiškinamasis dirbtinis intelektas galėtų padėti atpažįstant tokio tipo atakas.

3. DNT PAAIŠKINAMUMAS IR INTERPRETUOJAMUMAS

Paaiškinamasis dirbtinis intelektas (angl. *Explainable AI*, XAI) yra dar visai neseniai išpopuliarėjusi tyrimų sritis, kurios tikslas yra naudojantis įvairiais įrankiais ir principais suteikti skaidrumo labai sudėtingiems ir neaiškiems mašininio mokymosi (angl. *machine learning*) modeliams bei padėti dirbtinio intelekto modelių kūrėjams ir naudotojams sužinoti modelių sprendimų priėmimo motyvus. 2004-ais metais Michael van Lent, William Fisher ir Michael Mancuso pristatytas konceptas, skirtas paaiškinti dirbtinio intelekto kontroliuojamų dalių veiksmus simuliacinėse žaidimų programose [VFM04], dabar yra plačiai nagrinėjamas ir pritaikomas įvairiose srityse, tokiose kaip medicina, farmakologija, meteorologija bei astronomija, kur paaiškinimai gali būti naudojami sukurti naujas hipotezes ir geriau suprasti vykstančių procesų priežastingumą [Sam23]. Dirbtinių neuroninių tinklų paaiškinamumas leidžia suprasti priimtą sprendimą galutiniam vartotojui, taip sukuriant pasitikėjimą, jog sprendimas buvo priimtas nešališkai, remiantis faktais, o interpretuojamumas – susieti paaiškinimą su kontekstu.

Yra išskiriamos net kelios priežastys, kodėl yra svarbu mokėti paaiškinti ir interpretuoti dirbtinių neuroninių tinklų gaunamus rezultatus [SWM17]:

1. **Socialinis aspektas.** Sprendimus priiminėjant žmogui, jis remiasi tam tikrais samprotavimais ir priimtą sprendimą gali paaiškinti kitiems, taip užtikrindamas pasitikėjimą. To paties yra tikimasi ir kai sprendimas yra priimamas dirbtinių neuroninių tinklų.
2. **Sistemos patikimumas.** Naudojantis sistema kaip juodąja dėže, negalima ja pilnai pasitikėti. Vienas iš tai patvirtinančių pavyzdžių – dirbtiniu intelektu pagrįstos medicininės sistemos plaučių uždegimui nustatyti naudojimo atvejis [CLG+15]. Modelis buvo išmokęs, jog astma sergantys pacientai su širdies problemomis turi mažesnę tikimybę mirti nuo plaučių uždegimo nei šių ligų neturintys. Iš tiesų – pacientų su šiomis ligomis mirtingumo rodikliai buvo mažesni, bet tik dėl to, jog jiems nuolat buvo taikoma griežta medicininė priežiūra dėl turimų ligų. Medicinos daktarai iš karto žinotų, jog astma ir šidies ligos kaip tik yra faktoriai, neigiamai veikiantys gijimo procesą, tačiau modelis neturi jokių žinių apie šias ligas ir remiasi gauta koreliacija.
3. **Modelio tobulinimo galimybės.** Tam, kad galima būtų gerinti dirbtiniu intelektu paremtas sistemas, visų pirma reikėtų surasti jų silpnybes. Tai padaryti bei surasti šališkumus yra daug lengviau, kai yra žinoma, kodėl gautas sprendimas buvo priimtas. Kuo aiškiau yra ką modelis daro, tuo lengviau yra jį patobulinti.
4. **Papildomas mokymosi šaltinis.** Šiais laikais dirbtiniai neuroniniai tinklai yra mokomi su dideliu kiekiu pavyzdinių duomenų ir gali pastebėti šablonus duomenyse, kurių nepamatytų žmogus, galintis dirbti tik su ribotu kiekiu pavyzdžių. Mokant paaiškinti ir interpretuoti modelių rezultatus, galima būtų pabandyti išgauti šias žinias iš sistemos ir įgyti naujų įžvalgų. Tai ypač naudinga būtų fizikams, chemikams ir biologams, kurie nori identifikuoti naujus gamtos dėsnius, o ne apsiriboti rezultatu, gaunamu iš juodosios dėžės modelio.
5. **Atitikimas teisiniams reikalavimams.** Dirbtiniu intelektu paremtos sistemos yra vis plačiau naudojamos, todėl vis didesnis dėmesys yra skiriamas teisiniams aspektams (pvz., kas turėtų būti atsakingas, jei sistema pateikė blogą sprendimą ir dėl to kas nors nukentėjo). Taip

pat, galint paaiškinti modelio sprendimą, žmonėms, tiesiogiai veikiamiems dirbtinio intelekto sprendimo, būtų daug lengviau suprasti, ar nebuvo pažeistos jų teisės (pvz., sistemai nepatvirtinus paskolos išdavimo). Šie rūpesčiai paskatino Europos Sąjungą priimti naujus įstatymus, užtikrinančius „teisę į paaiškinimą“, pagal kuriuos vartotojas gali reikalauti paaiškinimo dėl algoritminio sprendimo, priimto dėl jos ar jo [GF17].

Paaiškinamumas ir interpretuojamumas tarpusavyje yra glaudžiai susiję ir kartu leidžia labiau pasitikėti modelio sprendimų priėmimo procesu [AAE+23]. Per kelis paskutinius metus buvo pasiūlyta įvairių metodų, kaip paaiškinti ir suprasti mašininio mokymosi modelius, kurie anksčiau buvo laikomi juodosiomis dėžėmis, pavyzdžiui – dirbtiniai neuroniniai tinklai, ir patikrinti jų sugeneruotus rezultatus. Yra išskiriami du XAI metodų tipai: lokalus (angl. *local*) ir globalus (angl. *global*). Naudojant lokalius metodus yra gaunami paaiškinimai tam tikram spėjimui, gautam su vienu duomenų rinkinio įrašu, kai tuo tarpu globalūs metodai suteikia įžvalgos apie modelio veikimą su visu duomenų rinkiniu [AZC+23]. Tokie metodai gali būti paremti trikdymu (angl. *perturbation-based*), gradientu (angl. *gradient-based*), pakaitalu (angl. *surrogate-based*), sklidimu (angl. *propagation-based*) ir kt [Sam23].

3.1. LIME

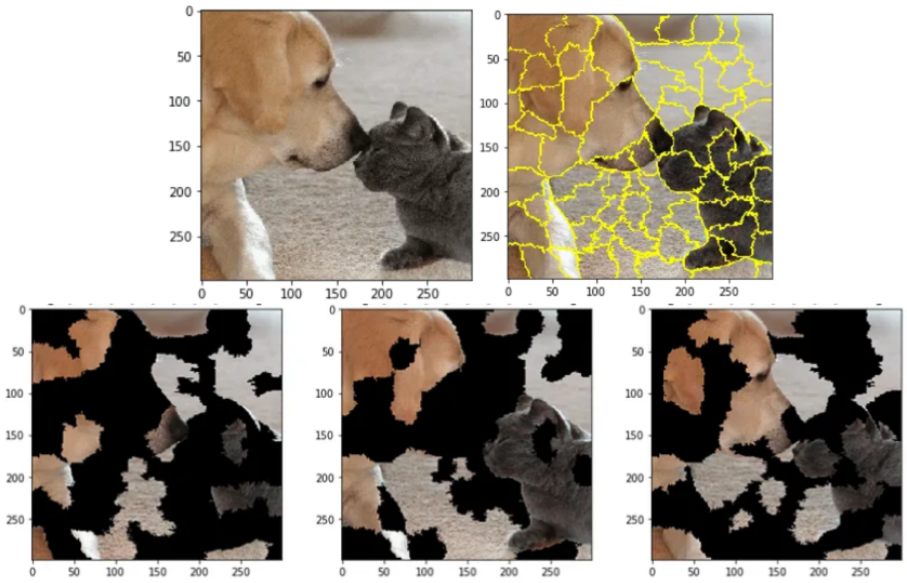
Vienas iš metodų, naudojamų paaiškinti KNT rezultatus yra vietiniai interpretuojami modelių agnostiniai paaiškinimai (angl. *Local Interpretable Model-Agnostic Explanations*, LIME) [RSG16]. Šis lokalus metodas gali paaiškinti bet kurio klasifikatoriaus ar regresoriaus prognozes atskiriems duomenų rinkinio įrašams.

LIME veikimo žingsniai vaizdams paaiškinti yra [GM21]:

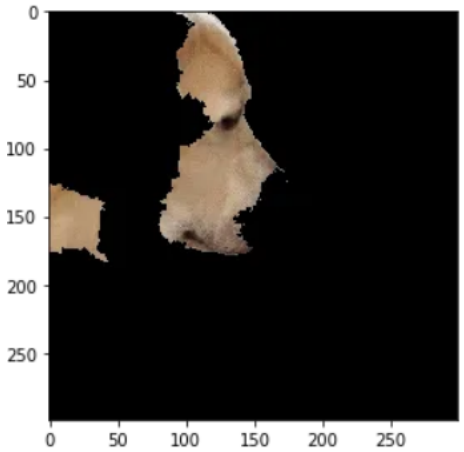
1. Pasirenkamas duomenų rinkinio pavyzdys (paveikslėlis), kuriam yra norima gauti neuroninio tinklo modelio prognozę.
2. Sukuriamas naujas pasirenkamo dydžio duomenų rinkinys, sudarytas iš originalaus paveikslėlio egzempliorių su perturbacijomis. Jos gaunamos vaizdą padalijus į superpikselius (besijungiančias vaizdo dalis, kurios turi panašias savybes) ir atsitiktinai išjungiant arba įjungiant juos (įtraukiant arba neįtraukiant juos į paveikslėlį). 11 paveikslėlyje yra pavaizduotas originalus pasirinktas paveikslėlis, suskirstymas į superpikselius ir perturbacijos, pritaikytos įtraukiant arba neįtraukiant superpikselius į paveikslėlį.
3. Gaunamos neuroninio tinklo prognozės sukurtam perturbuotų duomenų rinkiniui.
4. Sukurtam perturbuotų duomenų rinkiniui yra skaičiuojamos reikšmės, vadinamos svoriais, nusakančios, kiek panašūs duomenų rinkinio elementai yra į originalų paveikslėlį. Tai atliekama naudojant branduolio funkciją, kuri priskiria svorį kiekvienam pavyzdžiui, remiantis jo atstumu iki paaiškinamo pavyzdžio. Branduolio funkcija gali būti bet kokia funkcija, matuojanti panašumą, pavyzdžiui, Gauso branduolys.
5. Apmokomas lokalus, interpretuojamas modelis su sudarytu perturbuotu duomenų rinkiniu ir gautomis dirbtinio neuronino tinklo prognozėmis, naudojant praeitame žingsnyje gautus svorius. Šio modelio tikslas yra aproksimuoti sudėtingo modelio elgesį lokalioje srityje aplink pavyzdį, kurį norima paaiškinti. Lokalus modelis turėtų būti pakankamai paprastas, kad būtų

lengvai interpretuojamas, tačiau pakankamai tikslus, kad atspindėtų svarbius sudėtingo modelio požymius. Lokalių modelių pasirinkimas priklauso nuo problemos srities ir neuroninio tinklo modelio sudėtingumo. Dažnai naudojami modeliai yra tiesiniai modeliai, sprendimų medžiai ir taisyklėmis pagrįsti metodai.

6. Apmokytas lokalus modelis yra naudojamas neuronino tinklo modelio sprendimo paaiškinimams generuoti. Tai atliekama analizuojant lokalaus modelio koeficientus ir nustatant požymius, kurie labiausiai prisidėjo prie prognozės. Tokių požymių vaizdavimo pavyzdys yra pateiktas 12 pav.



11 pav. Originalus paveikslėlis (viršuje kairėje), originalus paveikslėlis, suskirstytas į superpikselius (viršuje dešinėje), trys originalaus paveikslėlio perturbacijos variantai, kuriuose yra įjungti ir išjungti skirtingi superpikseliai (apačioje).



12 pav. Originalaus paveikslėlio požymiai, kurie labiausiai lėmė dirbtinio neuroninio tinklo rezultatus.

LIME gali būti taikomas bet kokiam mašininio mokymosi modeliui, o tai leidžia naudoti šį metodą įvairiuose kontekstuose ir su įvairiais modeliais. Taip pat jis susitelkia į konkretaus duomenų pavyzdžio paaiškinimą, o ne bando paaiškinti viso modelio veikimą ir taip leidžia gauti

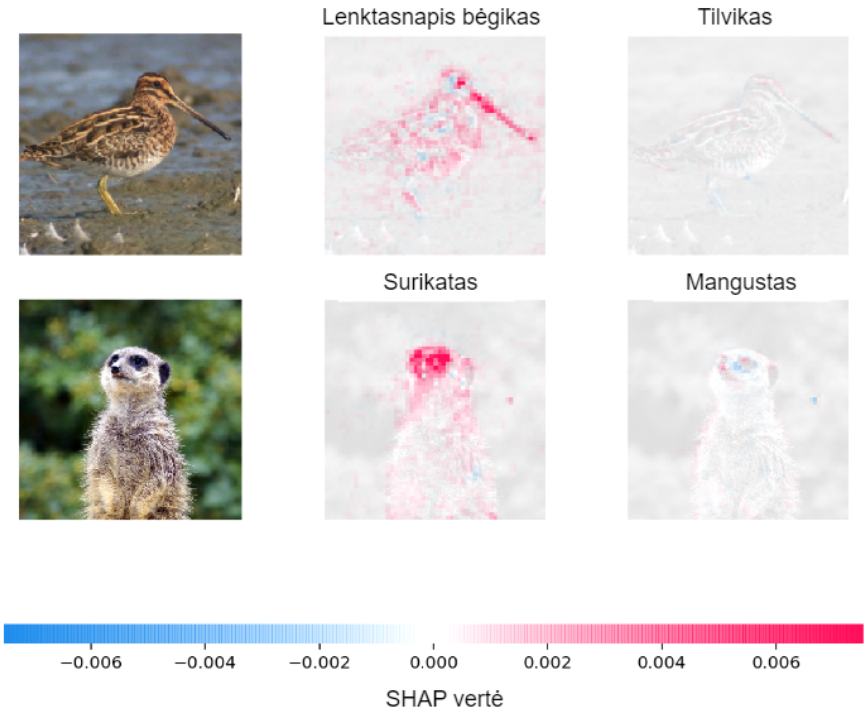
tikslesnius ir labiau suprantamus paaiškinimus žmogui. Padėdamas identifikuoti, kurios savybės labiausiai prisideda prie modelio prognozės, LIME metodas suteikia aiškumo ir prisideda prie modelių optimizavimo galimybių [RSG16].

3.2. SHAP

Kitas XAI metodas, naudojamas KNT rezultatų analizei yra SHapley priedų paaiškinimai (angl. *SHapley Additive exPlanations*, SHAP) [LL17]. SHAP yra naudojamas siekiant geriau suprasti, kaip modelis priima sprendimus ir nustatyti svarbiausius požymius, kurie daro didžiausią įtaką galutiniam rezultatui. Naudojant šį metodą yra analizuojami modelio rasti požymiai priskiriant jiems svarbos reikšmes, vadinamas SHAP reikšmėmis. Jos nurodo, kaip kiekvienas požymis prisideda prie modelio galutinės prognozės, kiekvieno požymio svarbą lyginant su kitas požymiais ir modelio priklausomybę nuo to, kaip požymiai sąveikauja tarpusavyje. SHAP metodas yra pakankamai naujas, tačiau jo praktinių aplikacijų jau galima matyti medicinos [NAA+24] ir finansų [BGM+20] srityse.

SHAP metodas yra paremtas kooperatyvine žaidimų teorija [Par06]. Shapley reikšmė yra būdas sąžiningai paskirstyti visų žaidėjų koalicijos sugeneruotą bendrą pelną pagal jų individualų indėlį. Mašininio mokymosi kontekste „žaidėjai“ yra įvesties duomenų požymiai (pvz., įvesties paveikslėlio pikseliai), o „pelnas“ yra modelio padaryta prognozė. Turint N požymių (paveikslėlių nagrinėjimo atveju – pikselių) egzistuoja 2^N įmanomų požymių poaibių. Apskaičiuoti kiekvienos savybės indėlį nėra įmanoma esant dideliam N skaičiui, todėl SHAP metodas neskaičiuoja tikrosios Shapley vertės kiekvienam požymiui, o naudoja mėginių ėmimą (angl. *sampling*) ir aproksimacijas, kad apskaičiuotų Shapley reikšmes [LL17]. Kiekvieno pikselio įnašas modelio sugeneruotai prognozei yra įvertinamas stebint modelio prognozių pokyčius, kai pikselis yra įtrauktas arba pašalintas iš paveikslėlio. Teigiamos SHAP reiškmės priskiriamos pikseliams, turintiems teigiamą įtaką prognozei, o neigiamos – pikseliams, kurie turi neigiamą įtaką, tai yra, dažniausiai priklauso kitai klasei. SHAP reikšmė yra lygi nuliui, jei pikselis nėra svarbus arba yra nenaudingas prognozei. Tai reiškia, kad SHAP metodas yra atsparus trūkstamiems duomenims ir yra užtikrinama, kad nereikalingi pikseliai neiškraipytų interpretacijos. SHAP reikšmės yra adityvios, tai reiškia, kad kiekvieno pikselio įnašas į galutinę prognozę gali būti apskaičiuotas nepriklausomai nuo vienas kito, po šių apskaičiavimų jie yra susumuojami. Ši savybė leidžia efektyviai apskaičiuoti SHAP reikšmes didelės apimties duomenų rinkiniuose [LL17].

SHAP metodo gautos reikšmės vizualizuojamos kaip šilumos žemėlapiai (angl. *heatmaps*), uždėti ant originalaus paveikslėlio, paryškinant kiekvieno regiono svarbą prognozėje. SHAP sugeneruoto vizualaus paaiškinimo pavyzdys yra pavaizduotas 13 paveikslėlyje. Šis metodas gali būti naudojamas su bet kokiu mašininio mokymosi modeliu ir yra veiksmingas analizuojant sudėtingus modelius ir sąveikas tarp požymių. Taip pat, SHAP gali generuoti atskiras vizualizacijas skirtingoms klasėms, parodant, kurios vaizdo dalys prisideda prie kiekvienos klasės prognozės. SHAP reikšmės nesikeičia, kai modelis pasikeičia, išskyrus atvejus, kai pasikeičia požymių įnašas prognozei. Tai reiškia, kad SHAP reikšmės pateikia nuoseklią modelio elgsenos interpretaciją, net jei pasikeičia modelio architektūra ar parametrai.

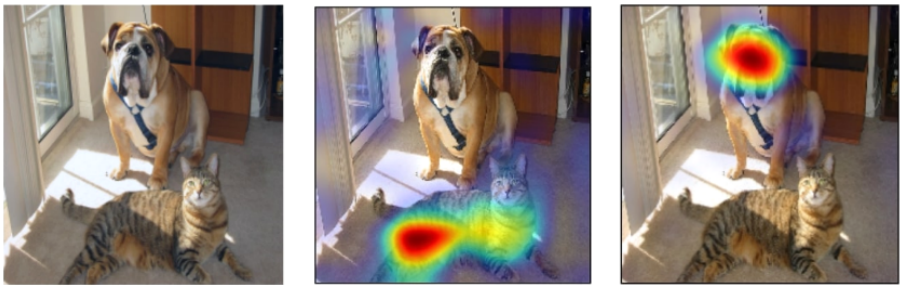


13 pav. VGG16 modelio septintam sluoksniui SHAP sugeneruota prognozės šilumos žemėlapių vizualizacija. Raudoni pikeliai reprezentuoja teigiamas SHAP vertes, tai yra, pikselius, darančius teigiamą įtaką prognozuojamai klasei. Mėlyni pikseliai reprezentuoja neigiamas SHAP vertes ir neigiamą įtaką prognozei. Pirmasis paveikslėlis suklasifikuotas į „lenktasis bėgikas“ klasę, antrasis paveikslėlis suklasifikuotas į „surikatas“ klasę [Lun18].

3.3. Grad-CAM

Grad-CAM (angl. *Gradient-weighted Class Activation Mapping*) yra metodas, naudojamas kompiuterinės regos srityje, ypač su KNT, siekiant pateikti vizualius paaiškinimus neuroninio tinklo sprendimams ir rezultatams [SCD+19]. Šis metodas sugeneruoja šilumos žemėlapius, kuriuose yra išryškinamos sritys įvesties vaizde, kurios turi didžiausią įtaką modelio prognozėms. Šis metodas gali sukurti atskiras vizualizacijas kiekvienai klasei, esančiai paveikslėlyje ir generuoja paaiškinimus bet kokiam KNT pagrįstam tinklui, nereikalaujant architektūrinių pakeitimų ar pakartotinio mokymo [SCD+19]. KNT klaidingo klasifikavimo atveju, šis metodas gali būti naudingas norint suprasti, kur ir kokios problemos yra neuroniniame tinkle, kaip jas išspręsti, padėti įsitikinti ar modelis fokusuojasi į tinkamus požymius klasifikavimo metu [SLS+22] [BAM+22] [ZCZ+22].

Grad-CAM metodo rezultatas yra originalus paveiklėlis su perdengtu šilumos žemėlapiu, paaiškinančiu, kurios sritys paveiklėlyje padarė didžiausią įtaką neuroninio tinklo klasifikacijos sprendimui. Grad-CAM sugeneruoto vizualaus paaiškinimo pavyzdys yra pavaizduotas 14 paveikslėlyje.

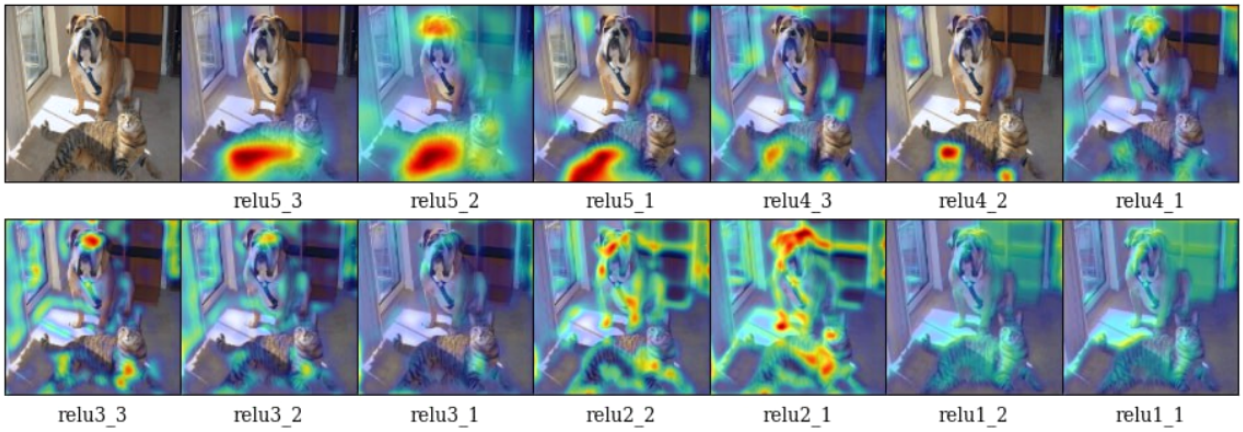


(a) Originalus paveiklėlis (b) Grad-CAM „Katė“ (c) Grad-CAM „Šuo“

14 pav. VGG16 modeliui Grad-CAM sugeneruoti šilumos žemėlapiai, paaiškinantys originaliame paveikslėlyje (a) rastas klases „katė“ (b) ir „šuo“ (c) [SCD+19].

Grad-CAM metodo veikimo žingsniai yra [SCD+19]:

1. Įvesties paveikslėlis yra praleidžiamas per pasirinktą KNT. Jam yra sugeneruojamos išvesties prognozės.
2. Yra identifikuojamas duominuojančios klasės balas (angl. *class score*). Šis balas paprastai yra gaunamas iš paskutiniojo sluoksnio prieš pritaikant *Softmax* funkciją.
3. Apskaičiuojami klasės balo gradientai pagal pasirinkto konvoliucinio sluoksnio požymių žemėlapius. Šie gradientai nurodo kiekvieno požymių žemėlapio svarbą klasės balui, tai yra, kaip kiekvienas rastas požymis, pvz., dryžiai ar akys, koreliuoja su norima vizualizuoti klase. Konvoliuciniai sluoksniai išlaiko erdvinę informaciją, kuri prarandama visiškai sujungtuose sluoksniuose, todėl Grad-CAM naudoja paskutiniojo konvoliucinio sluoksnio informaciją (15 pav.), kad priskirtų svarbos reikšmes kiekvienam neuronui (gilesnis konvoliucinis sluoksnis apima daugiau semantinių požymių).
4. Gautiems gradientams yra apskaičiuojami globalūs vidurkiai (angl. *global average pooling*) pagal erdvinius matmenis (aukštį ir plotį), taip gaunant svarbos svorius. Kiekvienas svoris atitinka konkretų požymių žemėlapį.
5. Paskutinio konvoliucinio sluoksnio požymių žemėlapiai yra pasveriami atitinkamais svarbos svoriais ir sumuojami. Taip gaunamas klasės šilumos žemėlapis.
6. Galutinis šilumos žemėlapis praleidžiamas per ReLU funkciją, kad būtų išlaikyti tik teigiamą įtaką norimai klasei darantys požymiai. Šis žingsnis paryškina sritis paveikslėlyje, teigiamai prisidedančias prie klasės balo. Neigiamą įtaką darantys priklauso kitoms klasėms. Neatlikus ReLU pritaikymo, šilumos žemėlapiai gali išryškinti ne tik norimą klasę.
7. Gautas šilumos žemėlapis padidinamas, kad atitiktų įvesties paveikslėlio dydį, ir uždedamas ant originalaus paveikslėlio, kad būtų vizualizuotos svarbios sritys atliktai klasifikacijai.



15 pav. Gard-CAM sugeneruoti šilumos žemėlapiai naudojant informaciją iš skirtingų konvoliucinių sluoksnių VGG16 architektūroje. Paveikslėlyje galima matyti, kad geriausios vizualizacijos yra gaunamos iš giliausio konvoliucinio sluoksnio tinkle (relu5_3 – penktojo konvoliucijos bloko trečiasis konvoliucijos sluoksnis), o lokalizacijos palaipsniui blogėja seklesniuose sluoksniuose [SCD+19].

Vaizdų klasifikavimo modelių kontekste, Grad-CAM sugeneruojamos vizualizacijos suteikia įžvalgas apie šių modelių veikimą, parodant, kad iš pažiūros nepagrįstos prognozės turi pagrįstus paaiškinimus, pranoksta ankstesnius modelių paaiškinimo metodus (c-MWP [ZBL+18], CAM [ZKL+16]) ILSVRC-15 konkurso silpnai prižiūrimos (angl. *weakly-supervised*) lokalizacijos užduotyje, yra atsparios priešiškoms perturbacijoms (angl. *adversarial perturbations*), yra labiau ištikimos pagrindiniam modeliui ir, identifikuojant duomenų rinkinio šališkumą, padeda pasiekti modelio generalizaciją [SCD+19].

3.4. Panašių tyrimų apžvalga

Tyrimų pavyzdžių, kuriuose XAI metodų pagalba yra tiriami priešiški puolimai ir vertinami dirbtinių neuroninių tinklų rezultatai, galima rasti ir mokslinėje literatūroje. Tyrimuose yra vykdomi eksperimentai naudojant skirtingas priešiškų puolimų technikas, įvairius XAI metodus ir taip pat yra bandoma pasiūlyti būdus, padedančius apsiginti nuo priešiškų puolimų.

Zuzanna Klawikowska ir kt. naudoja lokalius ir globalius XAI metodus atlikti analizei, ar įmanoma aptikti paveikslėlio užpuolimą [KMG20]. Tyrime yra atliekama kokybinė duomenų rinkinių savybių analizė siekiant nustatyti jų pažeidžiamumą užpuolimams. Lokaliai analizei pasirinktas LRP (angl. *Layer-wise Relevance Propagation*) XAI metodas, o globaliai analizei UMAP (angl. *Uniform Manifold Approximation and Projection*). Darbo metu tiriami dviejų tipų atakų poveikiai – FGSM ir vieno pikselio atakos. Siekiant įtikinamai pademonstruoti šių atakų poveikį ir padaryti išvadas, buvo naudojama duomenų bazė, sudaryta iš kačių ir šunų paveikslėlių. Abiejų užpuolimų atvejais prognozės tikslumas po užpuolimo ženkliai sumažėjo. Po FGSM atakos LRP sugeneruoto šilumos žemėlapio spalvos ir pikselių reikšmingumas prognozei stipriai pakito. Darbe siūlomas priešiškas modelio mokymas (angl. *adversarial training*), kaip sprendimas apsiginti nuo užpuolimų. Šio darbo rezultatai yra preliminarūs ir kokybiniai, pabrėžiantys giluminių modelių pažeidžiamumą tyčiniams įvesties trikdžiams.

Erzhena Tcydenova ir kt. darbe [TKL+21] LIME XAI metodas yra praktiškai pritaikomas pristatytame užpultų duomenų aptikimo modelyje. Siūloma sistema susideda iš dviejų fazių – inicializacijos ir aptikimo. Inicializacijos fazėje yra apmokoma įsibrovimo aptikimo sistema (angl. *Intrusion Detection System*, IDS), pagrįsta atramos vektorių mašinos (angl. *Support Vector Machine*, SVM) klasifikavimo modeliu. SVM modelis klasifikuoja gaunamus duomenis į normalius arba užpultus. Inicializacijos fazėje toliau yra sugeneruojami LIME paaiškinimai normaliems duomenims, kurie buvo naudojami IDS apmokymui. Paaiškinimas pateikia svarbiausių visų pasirinktų normalių duomenų įrašų savybių rinkinį. Aptikimo fazėje stebimas duomenų srautas pirmiausia analizuojamas IDS. Jei įvesties srautas klasifikuojamas kaip ataka, IDS siunčia įspėjimą. Jei jis klasifikuojamas kaip normalus, reikia dar vieno žingsnio, kad būtų patvirtintas IDS sprendimas. Šiame žingsnyje gaunamas naujas LIME paaiškinimų požymių rinkinys, naudojant inicializacijos fazėje sukurtą aiškintuvą (angl. *explainer*). Aiškintuvas pateikia N svarbiausių savybių, kuriomis remiantis buvo priimtas IDS sprendimas. Tada yra patikrinama, ar šios savybės yra inicializacijos fazėje LIME ištraukto normalių duomenų paaiškinimų požymių rinkinio dalis. Jei yra daugiau nei viena savybė, kuri nepriklauso rinkiniui inicializacijos metu sukurtam rinkiniui, šis įvesties srautas klasifikuojamas kaip ataka ir siunčiamas įspėjimas. Jei visos savybės priklauso rinkiniui, įvesties srautas klasifikuojamas kaip normalus. Darbe siūloma sistema sėkmingai aptinka priešiškas atakas realiu laiku, tačiau yra ribota binarinės klasifikacijos.

K.T. Yasas Mahima ir kt. taip pat aprašo tyrimą [MAP21], kuriame buvo vykdomi eksperimentai naudojant skirtingas XAI technikas, tokias kaip ryškumo žemėlapiai (angl. *saliency maps*), *DeepLift*, integruoti gradientai (angl. *integrated gradients*) ir SHAP siekiant įvertinti giliųjų neuroninių tinklų atsparumą priešiškoms atakoms. Buvo analizuojamas tinklo, apmokyto su CIFAR10 duomenų rinkiniu, sprendimo priėmimo procesas su paveikslėliais, kuriems priešiškos atakos buvo atliktos pasitelkiant FGSM, BIM (angl. *Basic Iterative Method*), PGD (angl. *Projected Gradient Descent*) ir paveikslėliais su įvairiais iškraipymais pasitelkiant spalvas, apšvietimą ir triukšmą. XAI metodai padėjo nustatyti, kurios vaizdų dalys yra svarbiausios modelio prognozėms – tyrimo rezultatai rodo, kad kai giliojo neuroninio tinklo modelis susiduria su priešiškais puolimais, būtinos pikselių savybės prognozėms priimti yra išsklaidytos per visą vaizdą, o ne specifiniame tikslo objekte. Tai suklaidina modelį ir jis pradeda naudoti per daug arba netinkamus pikselius sprendimui priimti. Kita vertus, paveikslėliuose, kurie buvo tik iškraipyti pasitelkiant spalvas, apšvietimą ir triukšmą, XAI metodai parodė, kad modelis dažniausiai vis tiek sugeba atpažinti svarbias savybes, nors ir su tam tikru tikslumo sumažėjimu. Šie iškraipymai daro mažesnį poveikį modelio sprendimų priėmimo procesui nei priešiškos atakos. Panaudojus priešišką mokymą arba duomenų transformacijų pagrindu veikiančią augmentaciją, modelis sugeba užfiksuoti atitinkamas pikselių savybes konkrečiame objekte arba sumažinti neigiamų pikselių savybių užfiksavimą. Tyrime yra pabrėžiama, jog priešiškas mokymas ir duomenų augmentacijos yra potencialūs gynybos mechanizmai siekiant padidinti modelio atsparumą priešiškiems puolimams.

4. PAAIŠKINAMUMO IR INTERPRETUOJAMUMO TYRIMAI

Praktinėje šio darbo dalyje yra siekiama ištirti dirbtinių neuroninių tinklų gautų rezultatų interpretavimo ir paaiškinamumo galimybes taikant paaiškinamojo dirbtinio intelekto metodus. Darbas suskirstytas į dvi tyrimų dalis. Pirmojoje dalyje yra atliekama vizuali LIME, SHAP ir Grad-CAM metodų sugeneruotų paaiškinimų analizė, lyginant juos originaliems bei triukšmu paveiktiems vaizdams. Šio etapo tikslas – nustatyti, kaip skirtingi metodai išskiria reikšmingus paveikslėlių regionus klasifikavimo metu bei kaip šie paaiškinimai kinta esant triukšmui. Antrojoje dalyje naudojant skaitines SHAP požymių svarbos vertes siekta nustatyti, ar paveikslėlis buvo paveiktas triukšmo, t. y., ar įmanoma aptikti priešišką ataką neturint tiesioginės informacijos apie paveikslėlio turinį. Šios dvi tyrimo dalys leidžia išsamiai įvertinti paaiškinamumo metodų efektyvumą ir potencialą taikant juos ne tik modelio interpretavimui, bet ir kaip papildomą saugumo priemonę prieš neuroninių tinklų pažeidžiamumą.

4.1. Vizualus metodų palyginimas

Tyrimo metu siekiant vizualiai įvertinti metodų gaunamus rezultatus buvo pasirinkti 3 viešai prieinami paveikslėliai iš `https://unsplash.com/` svetainės. Paveikslėliuose yra vaizduojami rudoji lapė, strutis ir paprika, jie yra vaizduojami 16 pav. Šie trys paveikslėliai buvo pasirinkti tik kaip pavyzdiniai, siekiant pademonstruoti gaunamus rezultatus, tad visus tyrimus galima būtų pakartoti ir su kitais paveikslėliais. Renkantis juos buvo siekiama išbandyti metodų veikimo galimybes su skirtingais požymiais – lapė stovi sulietame fone; stručio paveikslėlis margas, jame yra aiškūs fono elementai, tvora, šešėlis; paprikos vaizde fonas vienspalvis, aiškiai išsiskiria vaizduojamas objektas. Šie paveikslėlai buvo skirtingo dydžio, tad jų dydžiai buvo suvienodinti ir įgavo $(244, 244, 3)$ struktūrą, reiškiančią, jog jų aukštis ir plotis tapo po 244 pikslelius, sudarytus iš 3 spalvų (RGB). Šie paveikslėliai tada buvo paversti į masyvą ir iš anksto apdoroti naudojant $preprocess\_input$ funkciją iš $tensorflow.keras.applications.vgg16$ bibliotekos.

16 pav. Tyrimuose naudoti paveikslėliai.

Tyrimų metu paveikslėliai buvo klasifikuojami VGG16 neuroninio tinklo, kuris buvo iš anksto apmokytas su *ImageNet* duomenų rinkiniu, o tada gauti klasifikavimo rezultatai buvo paaiškinami naudojant LIME, SHAP ir Grad-CAM paaiškinamumo ir interpretuojamumo metodus. Vėliau paveikslėliai buvo užpuolami naudojant tikslines bei netikslines PGD atakas (2 skyrius). Tada triukšmu paveiktų paveikslėlių klasifikavimo rezultatai taip pat buvo bandomi paaiškinti naudojant

anksčiau minėtus metodus. Tyrimai buvo atliekami debesijoje esančioje užrašinėje *Google Colab*, leidžiančioje rašyti ir vykdyti *Python* kalba parašytą kodą per interneto naršyklę. Užrašinėje yra pasirinktas *Python 3* vykdymo aplinkos tipas ir aparatinės įrangos spartintuvas V100 GPU.

Darbo vykdymo metu leistas kodas gali būti pasiekiamas per nuorodą:

- `https://colab.research.google.com/drive/1rY6-uBhKbDhN8Ppj1eQ8vu6c2yhWYv-5?usp=sharing`

4.1.1. Priešiškas paveikslėlių užpuolimas

Tyrimų metu paveikslėliams buvo uždedamas triukšmas norint ištirti, kaip pasikeitė LIME, SHAP ir Grad-CAM metodų paaiškinimo rezultatai juos užpuolus triukšmu. Triukšmas buvo uždedamas dviem būdais – vykdant tikslinę ir netikslinę PGD atakas. Tikslinių atakų metu modelį buvo bandoma suklaidinti taip, jog jis pradėtų paveikslėlyje esantį objektą klasifikuoti kaip zebrą. Tikslinės ir netikslinės atakų veikimo principas yra labai panašus, tačiau skiriasi jų tikslas. Netikslinės atakos metu nėra svarbu, kokia neteisinga klasė bus priskirta paveikslėliui, o vykdant tikslinę ataką, klasės numeris yra nurodomas prieš jai įvykstant. Abu užpuolimai buvo vykdomi su vienodais parametrais:

- epsilon: 2;
- alpha: 0,5;
- iteracijų skaičius: 40.

Jų vykdymo metu paveikslėlis į funkciją buvo paduodamas kaip masyvas, sudarytas paveikslėlių suvienodinimo ir apdorojimo iš anksto metu, tačiau skyrėsi gradiento skaičiavimo būdas. Atakų metu buvo naudojamos skirtingos nuostolių funkcijos ir teigiamos arba neigiamos gradiento reikšmės, apskaičiuotos pagal nuostolių funkcijos reikšmę. Tikslinės atakos metu nuostoliai tarp norimos paveikslėlio klasės ir gautų rezultatų buvo skaičiuojami pasitelkiant kategorinę kryžminės entropijos funkciją. Gradiento skaičiavimui buvo imama neigiama šios funkcijos reikšmė, nes buvo siekiama, kad padidėtų tikimybė norimai klaidingai klasei, todėl reikėjo mažinti atstumą tarp spėjimo ir tikslo. Imant šią reikšmę be minuso, modelis stengtųsi mažinti nuostolius, t.y., tolti nuo norimos klasės. Gautas neigiamas gradientas buvo dauginamas iš alpha reikšmės, kuri nusako kiek vienos iteracijos metu paveikslėlis yra pakeičiamas pagal gradiento kryptį ir pridedamas prie turimo paveikslėlio, nukreipiant link norimos klasės. Tikslinei atakai įvykdyti buvo naudojamas šis kodas:

```
loss = -tf.keras.losses.categorical_crossentropy(target, preds)

grads = tape.gradient(loss, input_adv)
signed_grads = tf.sign(grads)
input_adv = input_adv + alpha * signed_grads
```

Netikslinės atakos metu buvo naudojama reta kategorinės kryžminės entropijos (angl. *categorical cross-entropy loss*) funkcija ir skaičiuojama nuostolių reikšmė tarp turimos klasės ir gautų spėjimų. Šiuo atveju tikslas buvo sumažinti tikimybę teisingai klasei, todėl teigiamas gradientas

buvo dauginamas iš alpha reikšmės ir pridedamas prie turimo paveikslėlio. Šiuo atveju kodas atrodė taip:

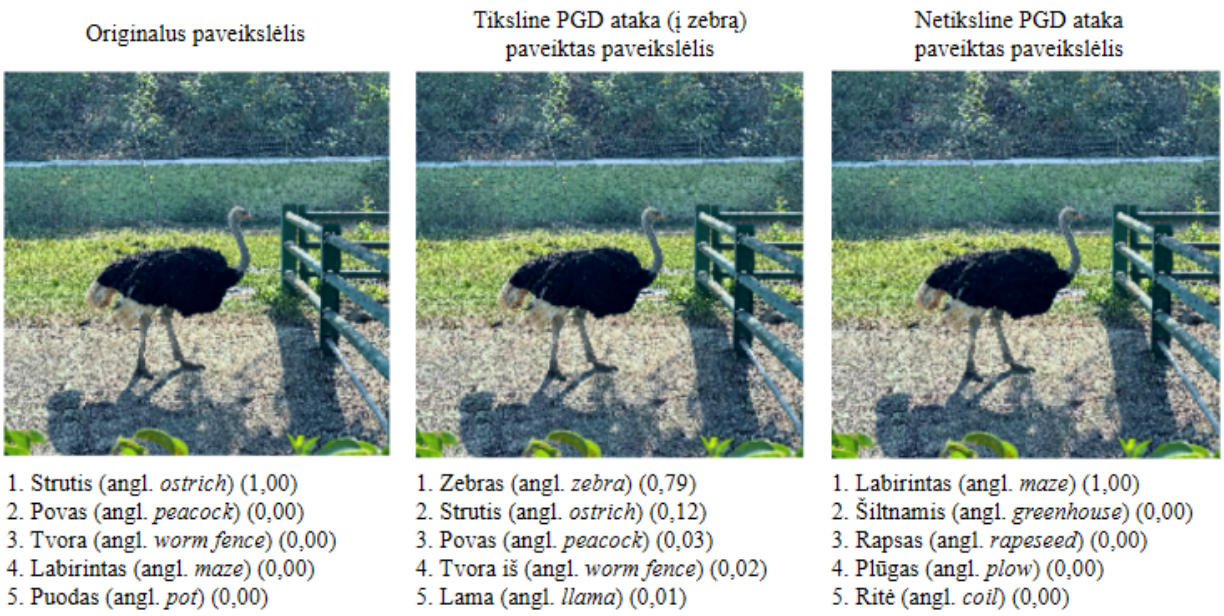
```
loss = tf.keras.losses.sparse_categorical_crossentropy(true_label, preds)

grads = tape.gradient(loss, input_adv)
signed_grads = tf.sign(grads)
input_adv = input_adv + alpha * signed_grads
```

17 paveikslėlyje yra pavaizduoti originalus lapės paveikslėlis bei atakų metu gauti vaizdai bei jų klasifikavimo rezultatai. Nagrinėjant šį bandymą užpulti paveikslėlį, kuriame pavaizduota rudoji lapė, matyti, jog šiame paveikslėlyje vizualiai labai ryškiai išsiskiria lapės spalva, dominuoja sulietas pikšvai mėlynas fonas, kuriame nėra konkrečių objektų. Tai lėmė, jog modelį buvo labai sunku suklaidinti vykdant tikslinę PGD ataką, nes įvykdžius 40 iteracijų, pasitikėjimo įvertis (angl. *confidence score*) tebuvo vos 0,55. Įvykdžius netikslinę ataką, neuroninis tinklas nusprendė, jog paveikslėlyje yra pavaizduotas kojotas. Šios klasės priklauso tai pačiai šuninių gyvūnų šeimai, todėl nėra stebėtina, jog turi panašius požymius.

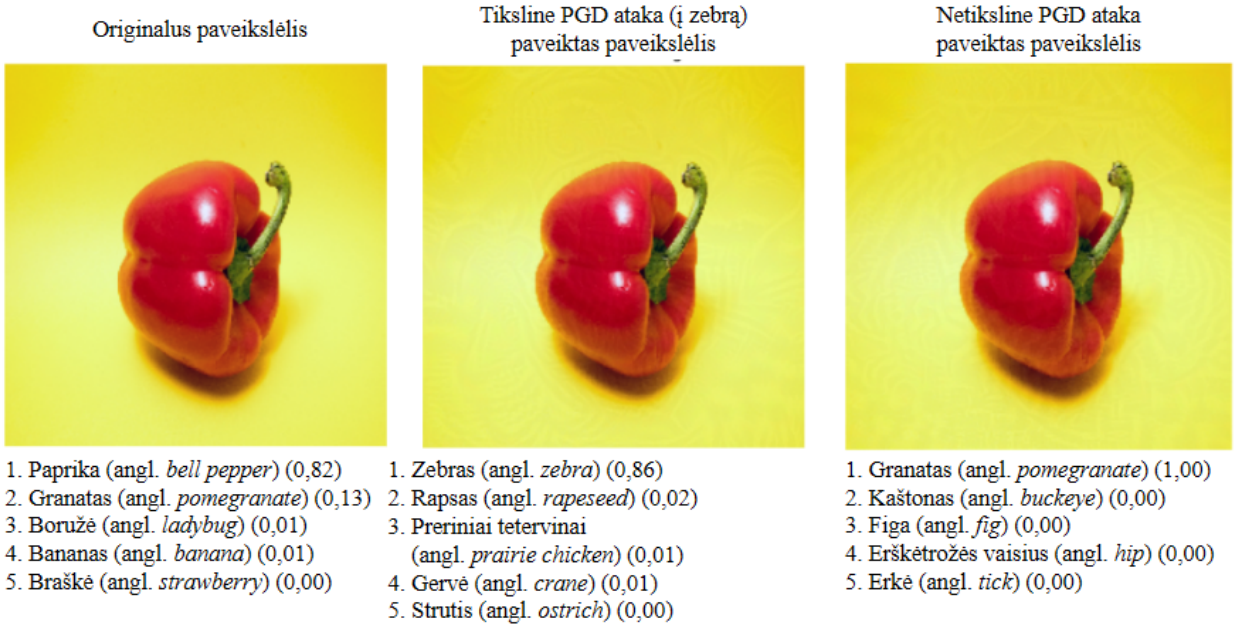
17 pav. Tyrimuose naudotas originalus lapės paveikslėlis ir tas pats paveikslėlis su uždėtu triukšmu bei jų klasifikavimo rezultatai.

Vizualiai vertinant originalų stručio paveikslėlį ir triukšmu užpultus vaizdus (18 pav.), nėra pastebimi jokie skirtumai tarp jų. Įvertinęs originalų paveikslėlį kaip strutį su tikslumo įverčiu vienas, neuroninis tinklas buvo suklaidintas abiejų atakų metu. Tikslinės PGD atakos metu pavyko strutį paversti į zebrą, o netikslinės atakos metu gautas rezultatas – labirintas – yra realus spalvoto, margo ir daug elementų, iš kurių vienas yra tvora, turinčio paveikslėlio pakeitimas.



18 pav. Tyrimuose naudotas originalus stručio paveikslėlis ir tas pats paveikslėlis su uždėtu triukšmu bei jų klasifikavimo rezultatai.

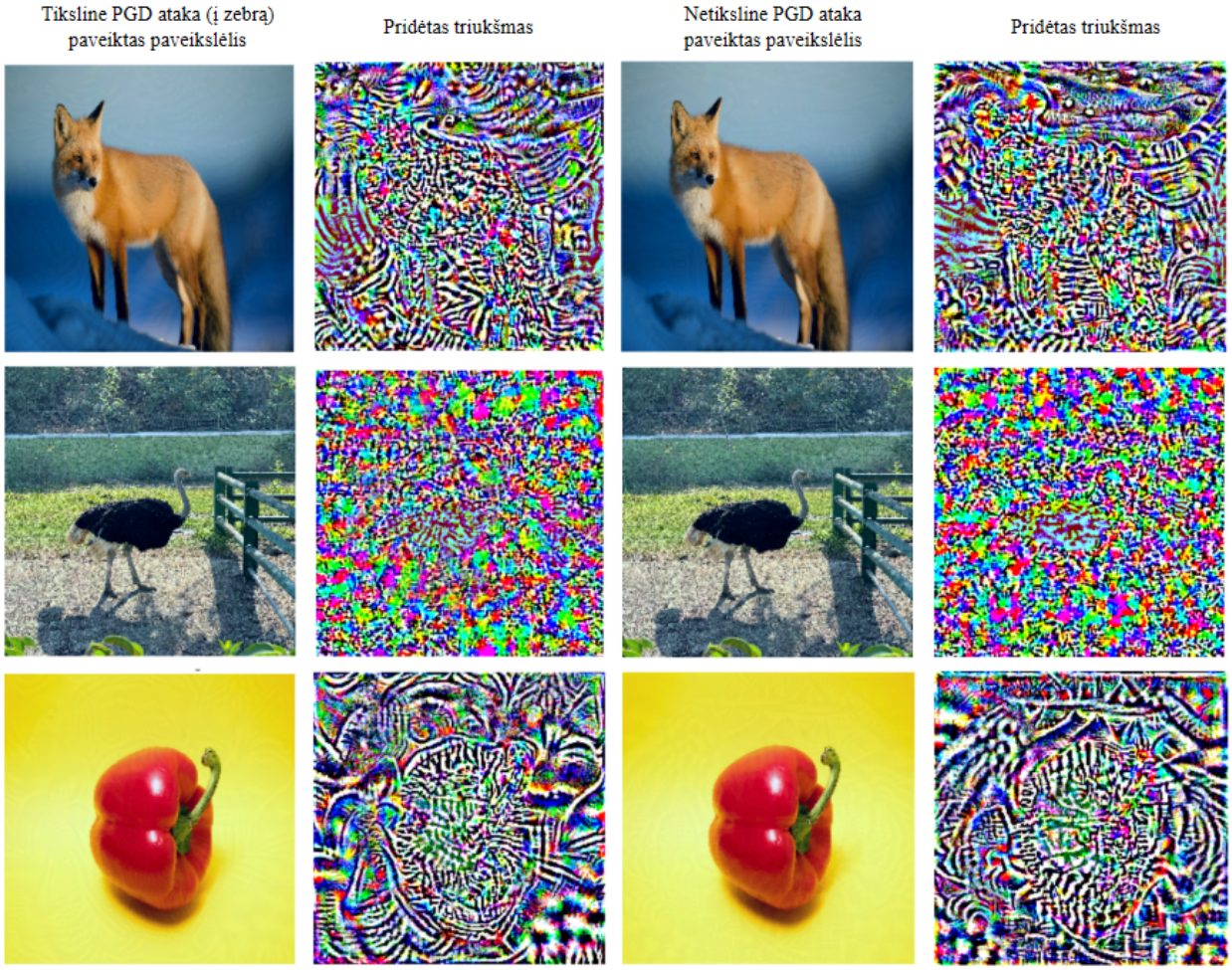
Pritaikant triukšmą paveikslėliui, kuriame yra pavaizduota paprika geltoname fone, vizualiai visuose paveikslėliuose vis dar taip pat galima atpažinti papriką, tačiau fone yra pastebimi šiokie tokie iškraipymai. Atakų metu gauti paveikslėliai bei originalaus ir užpultų paveikslėlių klasifikavimo rezultatai yra pavaizduoti 19 pav. Tiksline ataka paveiktas paveikslėlis buvo suklasifikuotas kaip zebras su pasitikėjimo įverčiu 0,86, o vykdant netikslinę ataką, modelis buvo visiškai įsitikinęs, jog paprika pavirto granatu.



19 pav. Tyrimuose naudotas originalus paprikos paveikslėlis ir tas pats paveikslėlis su uždėtu triukšmu bei jų klasifikavimo rezultatai.

20 paveikslėlyje yra pavaizduoti triukšmu paveikti paveikslėliai ir koks triukšmas buvo pa-

naudotas jiems gauti. Vertinant gautus klasifikavimo rezultatus yra pastebima, jog tikslinę ataką yra padaryti sunkiau ir galbūt neužtenka 40-ies iteracijų – bandant neuroninį tinklą suklaidinti uždedant triukšmą, jog paveikslėlyje yra pavaizduotas zebras, nei vienos iš atakų metu nepavyko pasiekti pasitikėjimo įverčio 1. Nepaisant to, jog neuroninis tinklas vis vien buvo suklaidintas ir paveikslėliuose pradėjo atpažinti zebro požymius, jis nebuvo tuo užtikrintas. Visiškai kitokie rezultatai buvo gauti vykdant netikslinę ataką. Kadangi klasė nebuvo apibrėžta iš anksto, pridedant triukšmą paveikslėlyje esantis objektas pavirto į vizualiai panašų kitos klasės egzempliorių. Įvykdžius šią ataką, modelis buvo įsitikinęs, jog paveikslėlyje yra pavaizduotas objektas, neatitinkantis vizualiai matomos jo klasės. Analizuojant gauto triukšmo vaizdinius lapės ir paprikos paveikslėliams yra pastebimi objekto, esančio originaliame paveikslėlyje, kontūrai, tačiau stručio paveikslėliui uždėtame triukšme jie nėra taip ryškiai matomi.



20 pav. Tiksline ir netiksline PGD atakų metu gauti paveikslėliai ir triukšmas, kuris buvo uždėtas ant originalaus paveikslėlio kiekvienos atakos metu.

4.1.2. Tyrimai naudojant LIME metodą

Atliekant tyrimus su paveikslėliu, LIME metodo (3.1 poskyris) funkcija suskirsto jį į segmentus. Tyrime pasirinkta naudoti numatytąjį (angl. *default*) greitojo poslinkio (angl. *quickshift*) segmentavimo algoritmą su parametrais:

- branduolio dydis (*kernel_size*) = 4;

- didžiausias atstumas (*max_dist*) = 200;
- santykis (*ratio*) = 0,2.

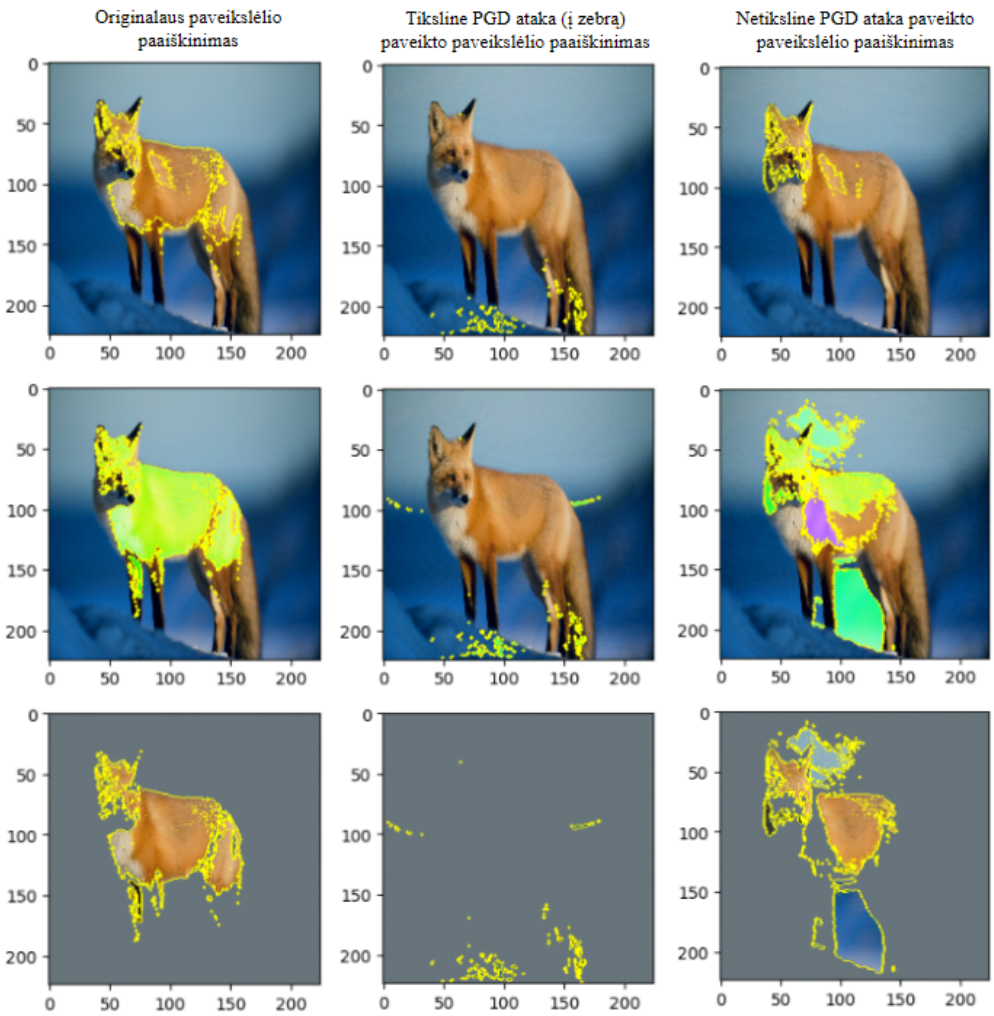
Šis algoritmas suskirstė visus paveikslėlius į skirtingą skaičių superpikselių. Po segmentacijos iš kiekvieno originalaus paveikslėlio yra sukuriamas 1000 egzempliorių turintis duomenų rinkinys. Šį duomenų rinkinį sudaro originalaus paveikslėlio kopijos su jam pritaikytomis perturbacijomis, kurios yra gaunamos atsitiktinai įtraukiant arba neįtraukiant turimus originalaus paveikslėlio segmentus. Sudarius naują duomenų rinkinį, skaičiuojamos naujos klasių tikimybės kiekvienam paveikslėliui naudojant tą patį VGG16 neuroninį tinklą. Sukurtam perturbuotų duomenų rinkiniui tada yra skaičiuojamos reikšmės, vadinamos svoriais, nusakančios kiek panašūs duomenų rinkinio elementai yra į originalų paveikslėlį. Tai atliekama naudojant branduolio funkciją, kuri priskiria svorį kiekvienam pavyzdžiui, remiantis jo atstumu iki paaiškinamo pavyzdžio. Tyrime pasirinkta naudoti numatytąją branduolio funkciją:

$$\pi_x(z) = \exp\left(-\frac{\text{dist}(x,z)^2}{\sigma^2}\right) \tag{5}$$

Čia $\pi_x(z)$ yra svoris, priskirtas perturbuotam paveikslėliui z, atsižvelgiant į pradinį paveikslėlį x, dist(x, z) yra atstumas tarp pradinio paveikslėlio x ir perturbuoto paveikslėlio z (kosinuso atstumas (angl. *cosine distance*), nusakantis dviejų vektorių skirtumą apskaičiuojant kampo tarp jų kosinusą), σ yra branduolio pločio parametras, kuris kontroliuoja svorių mažėjimą ir yra nustatomas remiantis perturbuotų paveikslėlių atstumų mediana.

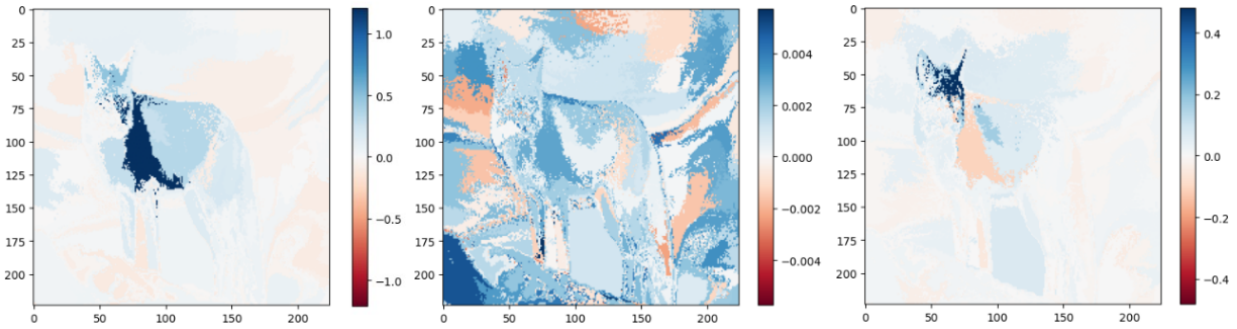
Tyrime naudojant LIME metodą buvo pasirinkta atvaizduoti po tris paaiškinimo pavyzdžius neužpultam ir užpultiems paveikslėliams. Pirmasis pavyzdys atvaizduoja penkis svarbiausius klasifikacijai požymius. Šie penki požymiai yra tik teigiamą įtaką darantys požymiai. Antrasis pavyzdys atvaizduoja dešimt svarbiausių požymių. Šie požymiai yra ir teigiamą ir neigiamą įtaką klasifikacijai suteikiantys požymiai. Trečiasis pavyzdys atvaizduoja dešimt svarbiausių teigiamų požymių ir uždengia likusį paveikslėlį, taip paryškinant įtaką klasifikacijai darančias zonas paveisklėlyje.

21 pav. galima matyti paveikslėlių paaiškinimus, gautus pasitelkiant LIME metodą. Nagrinėjant originalų lapės paveikslėlį galima pastebėti, jog modelis klasifikacijai pasitelkia tikrai prasmingas sritis – snukio kontūrus, ausis, lapės kūną, bei kojas. Visuose trijuose paaiškinimuose išlieka aiškus ir logiškas dėmesys gyvūnui bei yra išskiriami požymiai, pagal kuriuos žmogus taip pat priiminėtų sprendimą. Tikslinės PGD atakos paaiškinimuose pažymėtos sritys nebesutampa su semantiškai reikšmingais lapės bruožais – reikšmingi segmentai yra nebe lapės snukis, ausys ar uodega, o nedideli fono lopinėliai, kurie žmogui tikrai nepadėtų identifikuoti nei lapės, nei zebro. Modelis klasifikacijai pasitelkia vizualiai nesusijusią informaciją, tad ataka sėkmingai pakeitė ne tik klasifikaciją, bet ir modelio dėmesio žemėlapį. Paveikslėlio, kuriam buvo vykdyta netikslinė ataka paaiškinimuose, svarbiausiomis sritimis yra pažymimi tiek lapės, tiek fono segmentai. Į penkis labiausiai teigiamą įtaką sprendimui darančius požymius patenka snukis ir ausys, sąlyginai paaiškinantys, kodėl buvo priimtas sprendimas, kad paveikslėlyje yra pavaizduotas kojotas. Nagrinėjant dešimt svarbiausių tiek teigiamą, tiek neigiamą įtaką dariusių sričių, yra pastebima, jog kūno dalis buvo priskirta prie neigiamai sprendimą veikiančių gyvūno požymių, o fonas, esantis tarp lapės kojų ir virš jos, teigiamai prisidėjo prie neuroninio tinklo gauto klasifikacijos rezultato.



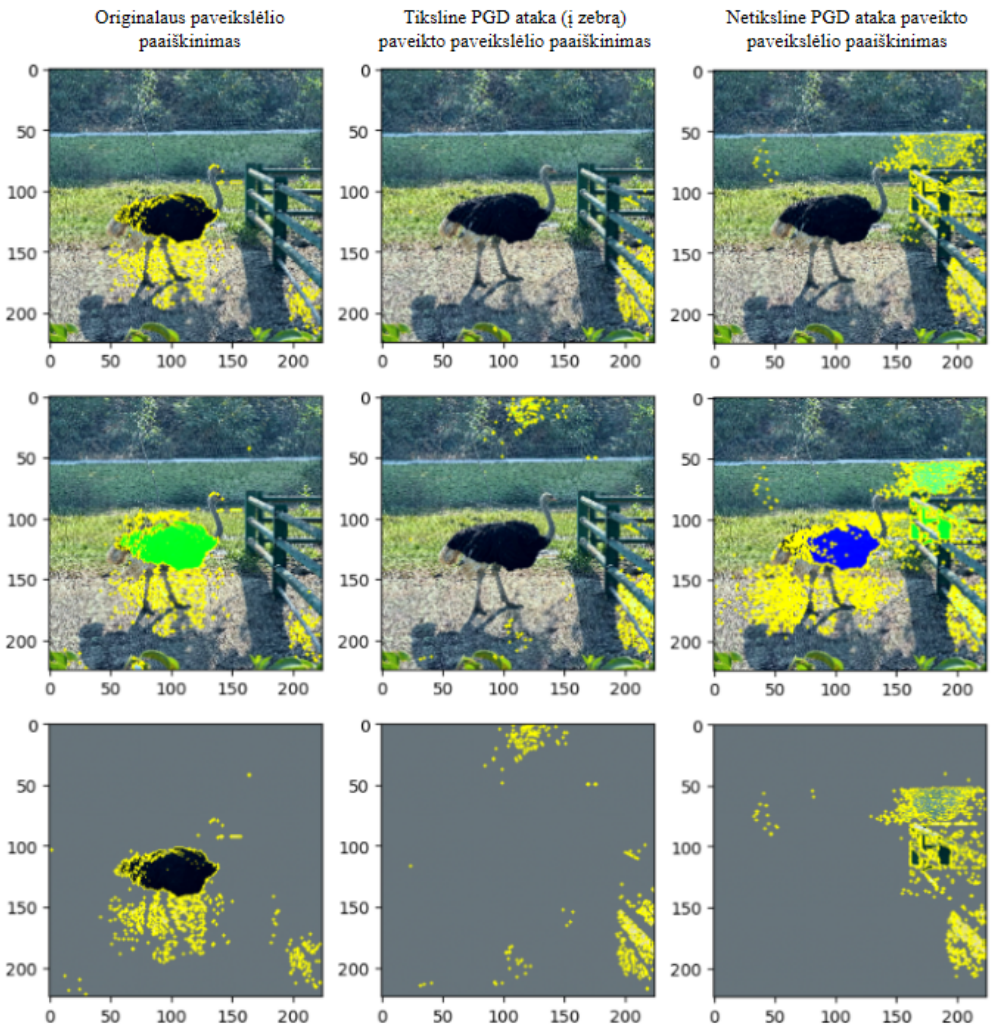
21 pav. LIME metodu sugeneruoti paaiškinimai originaliam lapės paveikslėliui (pirmas stulpelis), tiksline PGD ataka užpultam paveikslėliui (antras stulpelis) ir netiksline PGD ataka užpultam paveikslėliui (trečias stulpelis).

22 pav. yra pavaizduoti visi paveikslėlio segmentai ir jų įtaka (teigiama ir neigiama) klasifikavimo rezultatui. Svarbiausi segmentai paveikslėlyje yra nuspalvinami ryškiai mėlyna spalva, o labiausiai neigiamą įtaką klasifikacijai turintys – tamsiai raudona. Juose dar aiškiau įžvelgiami klasifikacijų metu pasitelktų požymių skirtumai. Neužpulto paveikslėlio (kairėje) fonas daro nežymią neigiamą įtaką lapės klasifikacijai, o lapės kūnas, ypač tamsiai mėlyna spalva paryškinta jo dalis daro stipriai teigiamą įtaką. Tai koreliuoja su požymiais, kuriuos išskirtų žmogus, bandantis atpažinti paveikslėlio subjektą. Tiksline PGD ataka užpulto paveikslėlio segmentų analizėje (viduryje), paveikslėlio segmentų chaotiškumas nesukelia daug pasitikėjimo rezultatais. Teigiamų sričių vertės yra mažos, išdėstytos po visą paveikslėlį ir persipynusios su neigiamą įtaką darančiais segmentais. Netiksline PGD ataka užpultame paveikslėlyje (dešinėje) pažymėta ausų sritis yra panaši į originalaus paveikslėlio paaiškinimą, o kūno dalis, originaliame paveikslėlyje pažymėta kaip svarbiausia teigiamą įtaką daranti paveikslėlio dalis, šiuo atveju neigiamai veikia klasifikavimo rezultatus. Lyginant verčių skalę taip pat yra pastebima, jog originalaus paveikslėlio mėlyni segmentai yra daug reikšmingesni nei triukšmu paveiktų paveikslėlių.



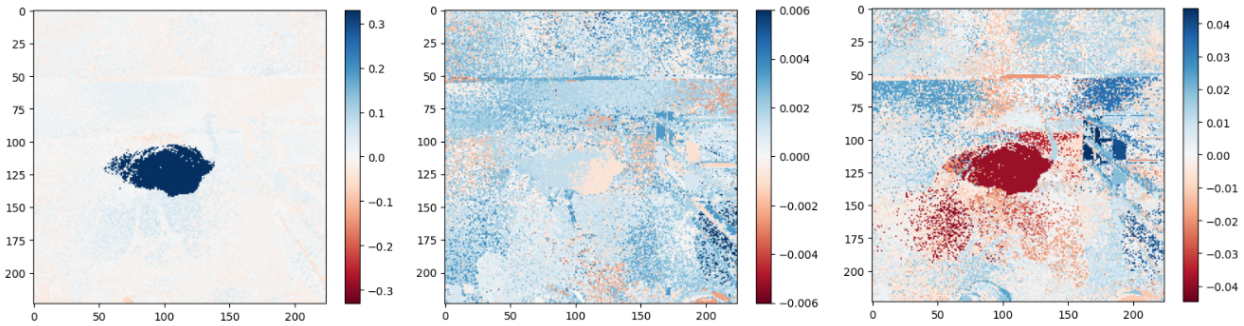
22 pav. Segmentų reikšmė originaliame (kairėje), tiksline PGD ataka užpultame (viduryje) ir netiksline ataka paveiktame (dešinėje) paveikslėliuose, kuriuose pavaizduota lapė.

23 pav. galima matyti gautus rezultatus pasitelkus LIME metodą originalaus ir užpultų stručio paveikslėlių paaiškinimui. Originaliame paveikslėlyje labiausiai klasifikavimo rezultatus lėmusi sritis yra svarbiausi stručio bruožai – jame yra aiškiai išskirtas kūnas ir plonų kojų forma. Fonas šalia stručio galimai sulaukia nedidelio dėmesio dėl artumo stručio kojoms ir padeda identifikuoti jo siluetą. Net ir vaizduojant dešimt požymių, neigiami segmentai nepasirodo – modelis užtikrintai identifikuoja objektą paveikslėlyje kaip strutį, vertindamas požymius, pagal kuriuos gyvūną atpažintų ir žmogus. Po tikslinės PGD atakos dėmesys nuo stručio nukrypsta – segmentai, turintys teigiamą įtaką klasifikacijai, atsiranda fone, ypač ties tvora ir žeme. Stručio kūnas nebepatenka į teigiamų požymių zonas, o tai rodo, kad ataka efektyviai nukreipė modelio dėmesį nuo originalaus objekto. Taip pat, kaip buvo pastebėta nagrinėjant tiksline PGD ataka užpultą lapės paveikslėlį, požymių, padedančių identifikuoti zebrą šiuose paaiškinimuose irgi nebuvo pastebėta. Netikslinės atakos atveju pasireiškia dar didesnis dėmesio išskaidymas – teigiami segmentai atsiranda fone, aplink tvorą ir žolę, o svarbiausias stručio požymis – kūnas – tampa pagrindiniu klasifikacijos sprendimą neigiamai veikiančiu požymiu. Trečiajame vaizde, kuriame matomi tik teigiami požymiai, nėra pažymėti jokie stručio segmentai, o tiesios tvoros linijos ir fono elementai tampa požymiais, lėmusiais modelio sprendimą, jog paveikslėlyje yra pavaizduotas labirintas.



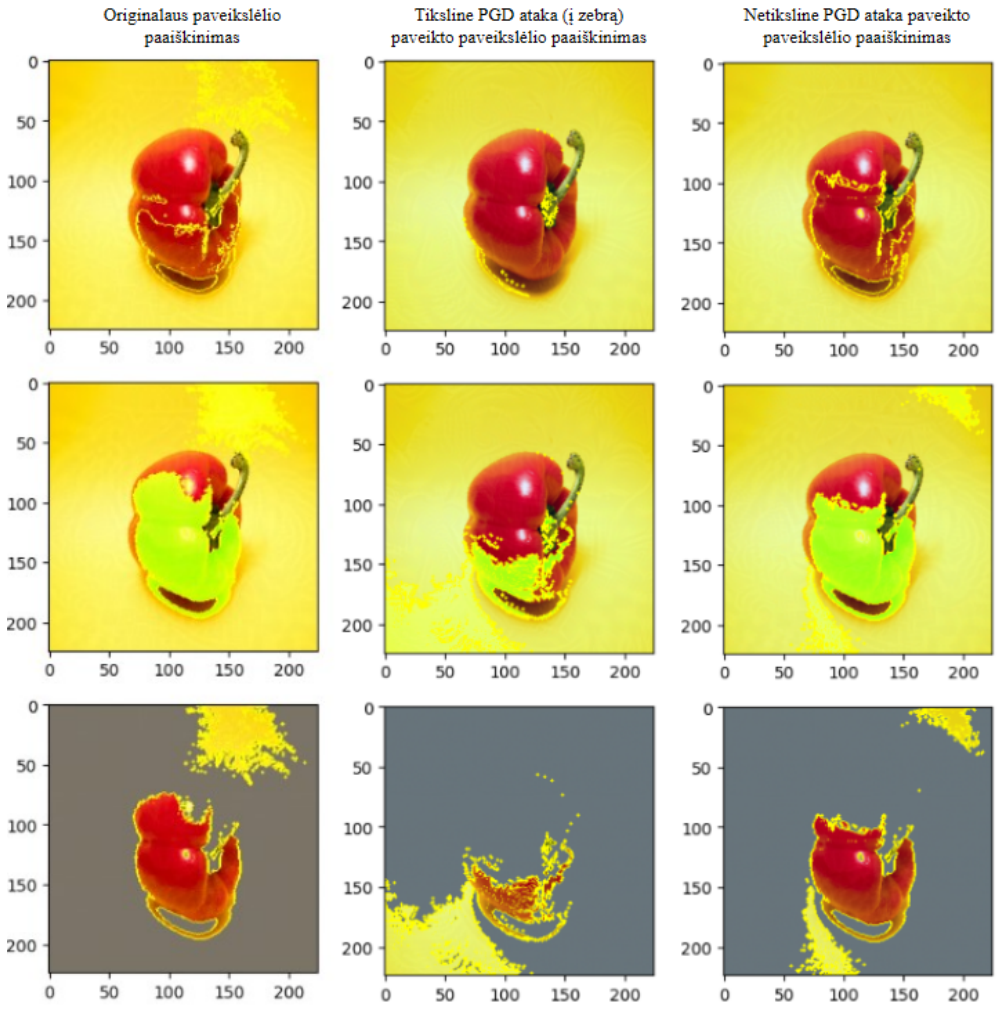
23 pav. LIME metodu sugeneruoti paaiškinimai originaliam lapės paveikslėliui (pirmas stulpelis), tiksline PGD ataka užpultam paveikslėliui (antras stulpelis) ir netiksline PGD ataka užpultam paveikslėliui (trečias stulpelis).

24 pav. atvaizduotame originalaus paveiklėlio paaiškinime (kairėje) yra išskiriamas stručio kūnas kaip itin svarbus požymis, visos kitos paveikslėlio dalys beveik nedaro įtakos modelio rezultatams. Tikslinės PGD atakos užpultame paveikslėlyje (viduryje) nebėra atpažįstamų požymių, visas paveikslėlis yra spalvų mišinys, o užpultame su netiksline PGD ataka (dešinėje) stručio kūnas yra paryškintas kaip labiausiai neigiamą įtaką darantis požymis. Lyginant šiuos rezultatus galima aiškiai matyti, kad triukšmo pridėjimas panaikino svarbiausius požymius paveikslėlio klasifikacijoje arba pavertė juos darančiais neigiamą įtaką klasifikacijos rezultatams. Taip pat, užpultuose paveikslėliuose yra pastebimas daug mažesnis spalvinių verčių intervalas. Tai reiškia, jog net ir mėlynai pažymėti paveikslėlio segmentai yra daug mažiau reikšmingi už originalaus paveikslėlio – uždėtas triukšmas suklaidino modelį.



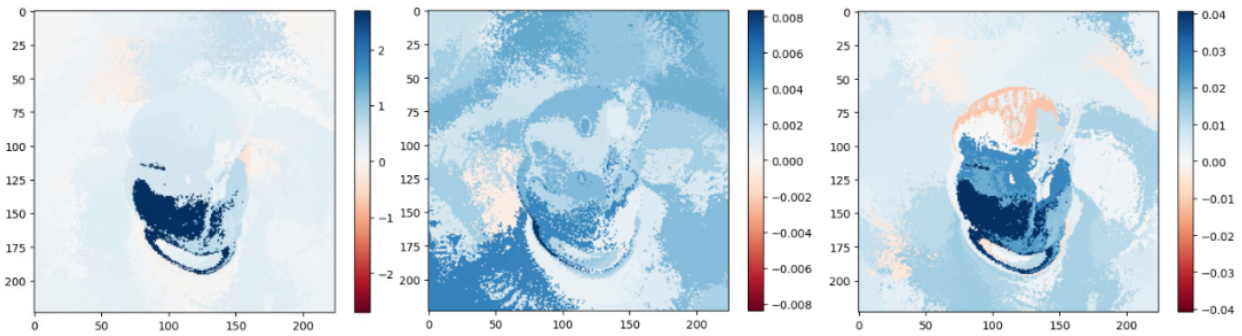
24 pav. Segmentų reikšmė originaliame (kairėje), tiksline PGD ataka užpultame (viduryje) ir netiksline ataka paveiktame (dešinėje) paveikslėliuose, kuriuose pavaizduotas strutis.

25 pav. pavaizduoti paprikos paveikslėlio rezultatai. Originaliame paveikslėlyje buvo sėkmingai identifikuota didžioji dalis paprikos – tarp penkių svarbiausių požymių (pirmoje eilutėje) patenka segmentai, išsidėstę aplink paprikos kraštus ir centrinę dalį, o dešimt požymių atskleidžia, kad beveik visa paprika turi teigiamą poveikį klasifikacijai, nėra pastebima jokių neigiamų zonų. Visuose originalaus paveikslėlio paaiškinimuose paprika ir jos kontūrai išlieka pagrindiniais požymiais, lėmusiais klasifikacijos rezultatus, tačiau yra pastebimas ir fono segmentas, turintis įtakos šiam sprendimui. Šis segmentas leidžia suprasti, kodėl gautas originalaus paveikslėlio klasifikavimo į papriką tikslumo įvertis nėra 1. Tiksline PGD ataka paveikto paveikslėlio paaiškimuose klasifikacijai svarbios sritys yra išskaidytos ir labiau pasiskirsčiusios fone, o paprikos objekte pažymėti segmentai yra labiau padriki bei nesudarantys nuoseklaus objekto vaizdo, dalis jų yra nutolę nuo pagrindinio objekto. Po netikslinės PGD atakos, paaiškinimuose taip pat dominuoja teigiami požymiai, tačiau svarbių segmentų ant pačios paprikos yra mažiau, jų taip pat yra randama aplink ją ar fone. Modelis šiuo atveju suklasifikavo paveikslėlyje esantį objektą kaip granatą, kuris yra vizualiai dydžiu ir spalva panašus objektas į papriką, tad šis paaiškinimas negalėtų padėti atpažinti, jog paveikslėlis buvo paveiktas triukšmu.



25 pav. LIME metodu sugeneruoti paaiškinimai originaliam paprikos paveikslėliui (pirmas stulpelis), tiksline PGD ataka užpultam paveikslėliui (antras stulpelis) ir netiksline PGD ataka užpultam paveikslėliui (trečias stulpelis).

26 pav. paaiškinimų segmentai įrodo gautų paaiškinimų rezultatus. Originaliame paveikslėlyje svarbiausias požymis yra paprikos dalis, o prieš tai išskirtas fono segmentas nėra stipriai reikšmingas. Reikšmingi tiksline PGD ataka užpulto paveikslėlio segmentai kaip ir praeituose pavyzdžiuose yra chaotiškai išsidėstę po visą paveikslėlį, o mažos jų skaitinės reikšmės nesukelia daug pasitikėjimo rezultatais. Neigiamų sričių šiame paveikslėlyje nėra pastebima daug, tad visas paveikslėlis yra įvardijamas kaip teigiamai veikiantis rezultatą, svarbiausiomis sritimis paryškinant vieną paprikos kraštą bei didelę dalį fono. Netiksline ataka užpultame paveikslėlyje, svarbiausias segmentas yra panašus į originalaus paveikslėlio, tačiau jo skaitiniai įverčiai yra daug mažesni. Taip pat yra pastebima, kad viršutinė paprikos dalis pradėjo neigiamai veikti rezultatą.



26 pav. Segmentų reikšmė originaliame (kairėje), tiksline PGD ataka užpultame (viduryje) ir netiksline ataka paveiktame (dešinėje) paveikslėliuose, kuriuose pavaizduota paprika.

Ištyrus paveikslėlius naudojant LIME metodą galima teigti, jog šis metodas yra pakankamai detalus išskiriant svarbiausias paveikslėlio dalis, darančias teigiamą ir neigiamą įtaką klasifikavimo rezultatams. Gautuose originalių paveikslėlių paaiškinimuose pagal segmentus dažniausiai galima įžiūrėti kontūrus ar bent jau bruožus objekto, kurį neuroninio tinklo modelis parenka kaip didžiausios tikimybės klasę. Tuos segmentus taip pat galima aiškiai matyti kaip didelę teigiamą reikšmę turinčias paveikslėlio dalis. Nagrinėjant triukšmu užpultų paveikslėlių segmentus, jie dažniau atrodo padriki, o verčių skalė būna ženkliai mažesnė. Segmentai, darantys teigiamą arbą neigiamą įtaką rezultatui nėra labai reikšmingi – neužpultuose paveikslėliuose jų reikšmės pasiekia daug didesnes vertes.

4.1.3. Tyrimai naudojant SHAP metodą

Antrasis tyrime naudotas paaiškinimo metodas yra SHAP. Naudojant šį metodą (3.2 poskyris), buvo analizuojami modelio rasti požymiai priskiriant jiems svarbos reikšmes, vadinamas SHAP reikšmėmis. Jos nurodo, kaip kiekvienas požymis prisideda prie modelio galutinės prognozės, kiekvieno požymio svarbą lyginant su kitas požymiais. Egistuoja keli SHAP reikšmių skaičiavimo būdai, šių tyrimų metu buvo naudotas segmentinis paaiškinimas (angl. *partition explainer*). Jo metu požymiai yra padalijami į grupes pagal jų sąryšį, o SHAP vertė tada yra skaičiuojama visiems grupės pikseliams, o ne kiekvienam pikseliui atskirai. Šio metodo rezultatas yra sugeneruotas šilumos žemėlapis kiekvienam imties paveikslėliui su teigiamą ir neigiamą įtaką klasifikacijai darančiais požymiais, kuris buvo panaudotas perdengiant aiškinamus paveikslėlius, kad būtų galima lengviau atpažinti įtakos sritis. Tyrime pasirinkta šilumos žemėlapiu perdengti sulietą (angl. *blurry*) originalų ir užpultus paveikslėlius ir SHAP metodu sugeneruoti keturių aukščiausios tikimybės klasių šilumos žemėlapių paaiškinimus. Paaiškinimų generacijai pasirinkta sudaryti naują duomenų rinkinį iš 1000 pavyzdžių ir paketo dydį lygų 50.

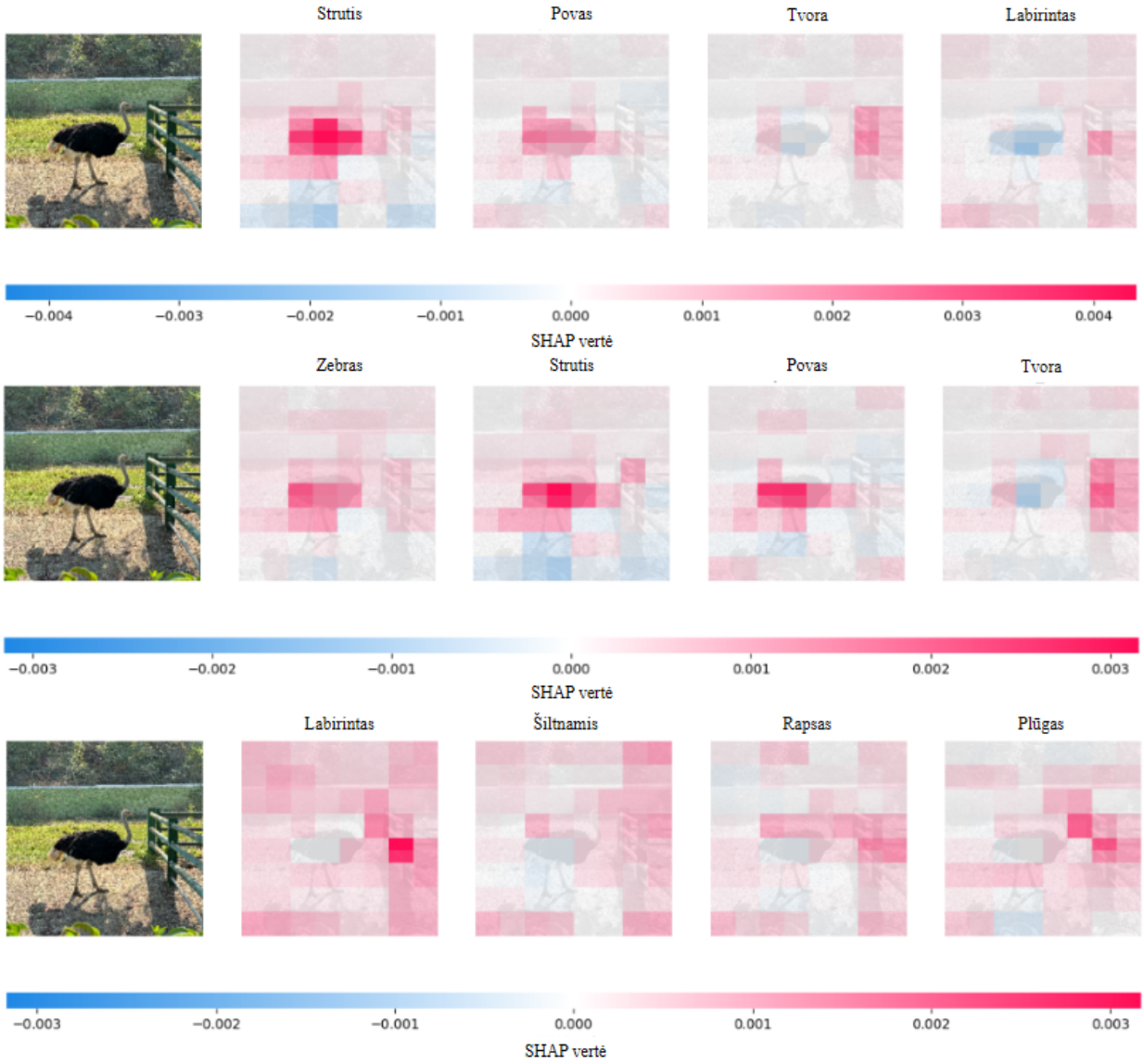
27 pav. yra pavaizduoti SHAP šilumos žemėlapiai, sudaryti originaliam ir užpultiems lapės paveikslėliams. Viršutinėje eilutėje originalus paveikslėlis aiškiai turi ryškius teigiamus požymius rudosios lapės klasėje. Šie požymiai apima galvą, nugarą, kojas – tai atpažįstami ir aiškūs lapės bruožai. Kitose klasėse nežymūs požymiai yra matomi panašiose vietose, tačiau SHAP vertės jiems yra ženkliai mažesnės nei rudosios lapės paveikslėlyje. Vidurinėje eilutėje, kurioje yra pavaizduoti tiksline PGD ataka paveikto paveikslėlio SHAP verčių žemėlapiai, ryškiausiai pažymėti segmentai

pirmose keturiose didžiausios tikimybės klasėse taip pat yra sutelkti aplink lapę, tačiau yra plačiau išsidėstę ir fone. Juose ryškiausios vertės visose keturiose klasėse yra aplink ausis, yra pastebimi ir mėlynos spalvos segmentai, neigiamai veikiantys klasifikavimo rezultatą. Netiksline PGD ataka paveikto paveikslėlio SHAP verčių žemėlapiai yra panašesni į originalaus, o ryškiausia rožinė spalva yra pastebima ten, kur modelis atpažino kojotą. Lyginant SHAP vertes originaliam ir triukšmu užpultiems paveikslėliams yra pastebima, kad originaliame paveikslėlyje išskirti segmentai yra reikšmingesni, o taip pat svarbūs segmentai ryškiai išskiriami tik didžiausios tikimybės klasėje, kai tuo tarpu triukšmu paveiktuose paveikslėliuose keturiose aukščiausios tikimybės klasėse buvo rasti reikšmingi požymiai.

27 pav. SHAP verčių šilumos žemėlapiai sugeneruoti originaliam (viršutinė eilutė) ir tiksline PGD ataka užpultam (vidurinė eilutė) ir netiksline PGD ataka užpultam (apatinė eilutė) rudosios lapės paveikslėliams.

28 pav. yra pavaizduoti SHAP šilumos žemėlapiai sugeneruoti stručio paveikslėliui. Originalaus paveikslėlio paaiškinimuose didžiausios tikimybės klasė turi daugiausiai svarbių požymių – paveisklėlyje ryškiausia rožine spalva yra pažymėti stručio kūnas, kojos, kaklas ir spalva. Blausiai

mėlynai žymimas aplinkinis stručio fonas: žemė, tvora – tai zonos darančios neigiamą įtaką klasifikavimo rezultatui. Likusiuose trijuose originalaus paveikslėlio paaiškinimuose spalvos yra ne tokios ryškios – tai reiškia, kad nors šios klasės buvo tarp keturių aukščiausių klasių tikimybių, jose reikšmingų pikselių, stipriai lemiančių būtent tą klasę, yra mažiau. Vidurinėje, tiksline PGD ataka užpulto paveikslėlio eilutėje aukčiausios tikimybės klasė yra zebras. Paaiškinimas išskiria stručio kūną kaip reikšmingai teigiamą zoną paaiškinimui, tačiau reikšmingi segmentai yra pastebimi ir fone bei paveikslėlyje matomos tvoros zonoje. Antrame paveikslėlyje modelis atpažino strutį, tačiau nepaisant ryškesnių rožinių segmentų nei didžiausios tikimybės klasėje, paaiškinimo metu atsirado ir ryškesnių klasifikaciją neigiamai veikiančių segmentų. Likusiose dviejose klasėse taip pat yra šiek tiek rausvos spalvos ir mėlynų segmentų. Apatinėje eilutėje, kurioje yra pavaizduoti po netikslinės PGD atakos gauti SHAP šilumos žemėlapiai, pimajame paveikslėlyje yra aiškiai matomi padriki rožinės spalvos segmentai. Visuose gautuose paaiškinimuose dominuoja teigiamai veikiančios stritys, tačiau jos neapibrėžia kokio nors konkretaus požymio, o yra pasiskirsčiusios viso paveikslėlio ribose. Triukšmu paveiktų paveikslėlių paaiškinimuose taip pat yra pastebimos ir mažesnės SHAP vertės, tad galima teigti, jog taip pat, kaip buvo pastebėta naudojant LIME metodą, paaiškinimų metu svarbiausi požymiai yra pastebimi aiškinant originalų paveikslėlį.



28 pav. SHAP verčių šilumos žemėlapiai sugeneruoti originaliam (viršutinė eilutė) ir tiksline PGD ataka užpultam (vidurinė eilutė) ir netiksline PGD ataka užpultam (apatinė eilutė) stručio paveikslėliams.

29 pav. matyti su paprikos paveikslėliui gauti SHAP šilumos žemėlapiai. Viršutinėje eilutėje esančiuose originalaus paveikslėlio paaiškinimuose aiškiai matomas paprikos klasės dominavimas – ryški rožinė spalva visoje paprikos srityje. Kitose klasėse požymiai yra nežymūs, jų SHAP vertės yra ženkliai mažesnės nei paprikos paaiškinime. Vidurinėje eilutėje, kurioje yra pateikti tiksline PGD ataka užpulto paveikslėlio paaiškinimai, visose klasėse yra pažymėtos tiek teigiamos tiek neigiamos stritys, chaotiškai išsidėsčiuosios viso paveikslėlio ribose. Šiuose paaiškinimuose pastebima, kad sprendimui reikšmingiausi požymiai buvo pastebėti geltoname fone, o pats paprikos objektas labiau neigiamai veikė klasifikacijos rezultatus. Apatinėje eilutėje, netiksliniu PGD metodu užpulto paveikslėlio SHAP šilumos žemėlapiuose matome daugiau panašumų į originalaus paveikslėlio paaiškinimą – čia didžiausios tikimybės klasėse taip pat yra pažymėta didžioji dalis paprikos, tačiau prie reikšmingų segmentų yra įtraukiama daugiau fono.

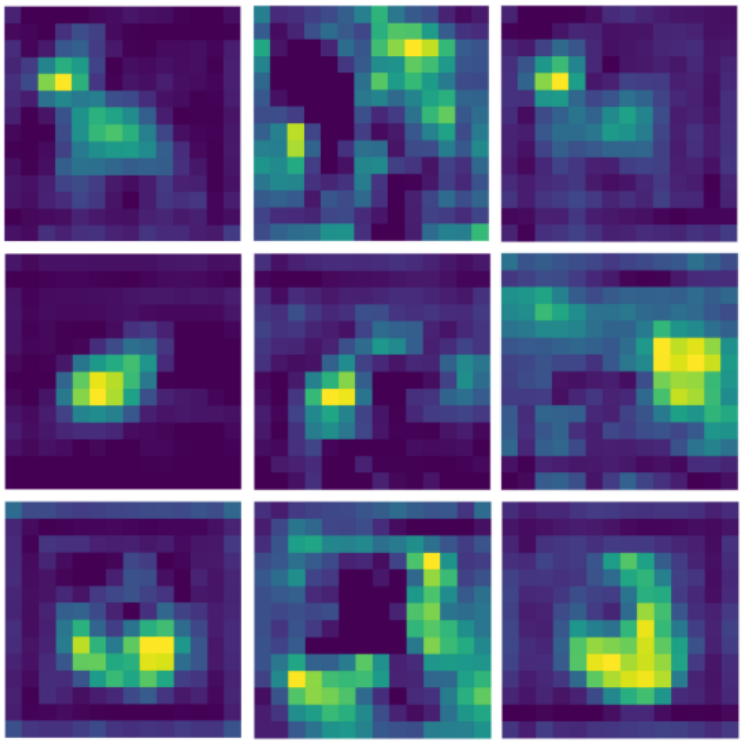
29 pav. SHAP verčių šilumos žemėlapiai sugeneruoti originaliam (viršutinė eilutė) ir tiksline PGD ataka užpultam (vidurinė eilutė) ir netiksline PGD ataka užpultam (apatinė eilutė) paprikos paveikslėliams.

Naudojant SHAP paaiškinamumo ir interpretuojamumo metodą, galima matyti paveikslėlio sritis, kurios daro teigiamą (rožinės) arba neigiamą (mėlynos) įtaką klasifikacijos rezultatams. Yra pastebima, jog originalaus paveikslėlio paaiškinimuose, ryškiausi segmentai yra matomi didžiausios tikimybės klasėje, tuo tarpu triukšmu užpultuose paveikslėliuose SHAP vertės yra panašios visuose keturiuose didžiausias tikimybes turinčių klasių paaiškinimuose. Originalūs paveikslėliai taip pat pasiekė didesnes SHAP vertes, tad juose pažymėti segmentai yra reikšmingesni nei užpultų paveikslėlių paaiškinimuose.

4.1.4. Tyrimai naudojant Grad-CAM metodą

Naudojant Grad-CAM metodą (3.3 poskyris), yra sugeneruojamas šilumos žemėlapis, kuriame yra išryškinama paveikslėlio sritis, turinti didžiausią įtaką modelio prognozėms. Tyrimų metu šie šilumos žemėlapiai buvo sugeneruoti originaliems ir triukšmu užpultiems paveikslėliams (30 pav.) ir uždėti ant įvesties paveikslėlių, kad galima būtų aiškiai matyti, kurios paveikslėlio

dalys prie klasifikavimo rezultatų prisidėjo daugiausiai. Gauti rezultatai yra pavaizduoti 31 paveikslėlyje, kur matyti, kad gauti šilumos žemėlapiai yra pakankamai abstraktūs, išskiriantys pačias paprasčiausias formas paveikslėliuose. Geltona spalva žymi pačius reikšmingiausius klasifikacijai pikselius, tuo tarpu tamsiai violetinė spalva žymi nereikšmingiausius. Šiuos šilumos žemėlapius galima sugeneruoti kiekvienam neuronino tinklo sluoksniui, tačiau populiariausiai naudojamas yra paskutinis konvoliucinis sluoksnis, turintis daugiausiai semantinių požymių. Tyrimų metu taip pat buvo naudojamas šis sluoksnis, VGG16 tinkle vadinamas $block5\_conv3$.



30 pav. Šilumos žemėlapiai sugeneruoti iš originalių (pirmasis stulpelis), užpultų su tiksline PGD ataka (vidurinis stulpelis) ir užpultų su netiksline PGD (trečiasis stulpelis) paveikslėlių, naudotų tyrimų metu.

31 pav. Šilumos žemėlapiai perdengti ant originalių (pirmasis stulpelis), užpultų su tiksline PGD ataka (vidurinis stulpelis) ir netiksline PGD ataka (trečiasis stulpelis) gautų paveikslėlių, naudotų tyrimų metu.

Analizuojant gautus rezultatus galima matyti, jog šis metodas yra abstrakčiausias iš nagrinėtų. Netiksline PGD ataka užpultame lapės paveikslėlyje buvo pažymėtos panašios gyvūno sritys kaip ir originaliame, tačiau nėra stebėtina, kad to pačios šeimos gyvūnai turi panašius požymius, lemiančius rezultatus. Šilumos žemėlapiai originaliam ir netiksline ataka užpultam paprikos paveikslėliams taip pat yra panašūs ir pažymi objekto vietas, tinkančias abiems klasifikavimo rezultatams. Tai papildomos informacijos nesuteikia, o tik patvirtina, jog pridedant triukšmą paveikslėlyje esantis objektas pavirto į vizualiai panašų kitos klasės egzempliorių. Galima pastebėti, jog šilumos žemėlapiuose paveikslėliams, užpultiems tikslinės PGD atakos metu, pažymėtos sritys yra išsidėsčiusios po visą paveikslėlį. Tai patvirtina, kad modelis nebesugebėjo identifikuoti pagrindinių požymių, įrodančių klasifikacijos rezultatus. Taigi, Grad-CAM metodas gali suteikti įžvalgų apie modelio veikimą ir padėti suprasti, kurios paveikslėlio vietos prisideda prie gautų rezultatų reikšmų daugiausiai, tačiau yra per daug abstraktus, kad padėtų tiksliau išanalizuoti triukšmo padarytą įtaką klasifikacijos rezultatams.

4.2. Skaitinių reikšmių įvertinimas

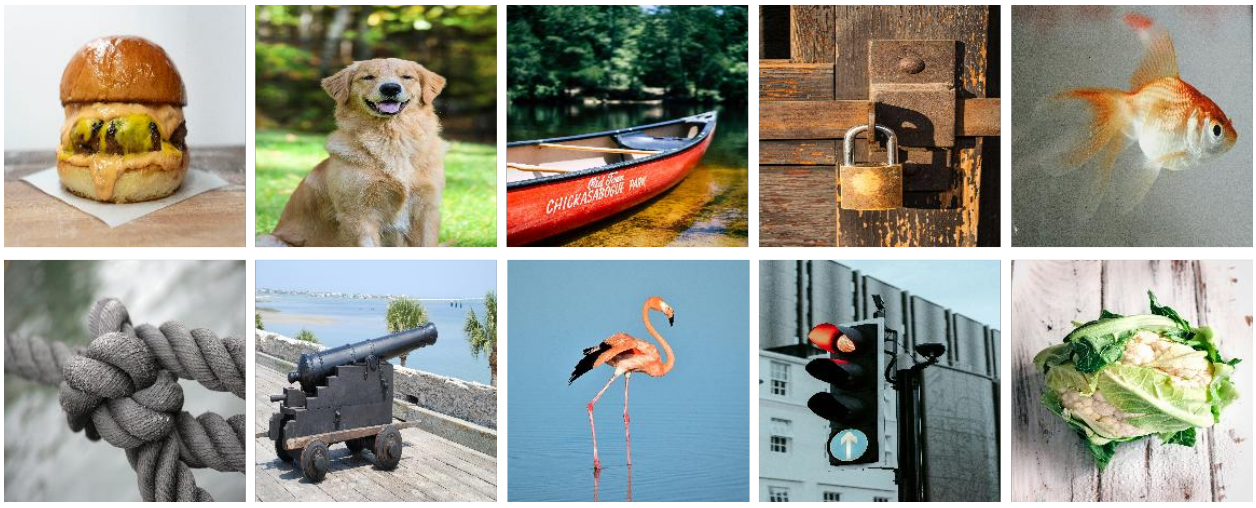
Analizuojant LIME, SHAP ir Grad-CAM metodų vaizdinius rezultatus galima vizualiai įvertinti, kiek atvaizduoti svarbiausi požymiai atitinka objekto, kurį matome paveikslėlyje, savybes, tačiau pastebėti šablonus ar konkrečius skirtumus tarp originalių ir priešiškai užpultų paveikslėlių taip pat gali padėti skaitinės metodų naudojimo metu gaunamos vertės. Tyrimo metu buvo pasirinkta naudoti SHAP metodu gaunamas skaitines reikšmes, apskaičiuotas kiekvienam pikseliui. SHAP

metodas yra paremtas kooperatyvine žaidimų teorija, kur gaunamos Shapley reikšmės yra būdas sąžiningai paskirstyti visų žaidėjų koalicijos sugeneruotą bendrą pelną pagal jų individualų indėlį. Šiuo atveju buvo bandoma ištirti, kokią įtaką įvesties duomenų pikseliai turi gaunamam rezultatui ir nustatyti, ar tos vertės gali padėti atpažinti, jog paveikslėlis buvo paveiktas triukšmu. Tyrimas buvo vykdomas su 50 paveikslėlių. Iš pradžių buvo suskaičiuotos kiekvieno iš jų pikselių svarbumo vertės ir įvairios metrikos joms. Tada paveikslėliams buvo uždėtas triukšmas naudojant tikslinę ir netikslinę PGD metodo atakas bei gauti skaitiniai rezultatai jiems. Iš originalių ir triukšmu užpultų paveikslėlių metrikų buvo sudarytas naujas duomenų rinkinys, kuris buvo naudojamas apmokyti mašininio mokymosi algoritmą, ar paveikslėlis yra paveiktas triukšmu. Šios tyrimo dalies vykdymo metu leistas kodas gali būti pasiekiamas per nuorodą:

- `https://colab.research.google.com/drive/13AEXk8xwf7b-jQHfOh3FcteDRrPXgcLM?usp=sharing`

4.2.1. Duomenų rinkinys ir SHAP reikšmių skaičiavimas

Duomenų rinkiniui sudaryti buvo pasirinkti 50 viešai prieinamų paveikslėlių iš `https://unsplash.com/` svetainės. Originalūs paveikslėliai buvo skirtingo dydžio, tad jų dydžiai buvo suvienodinti ir įgavo (244, 244, 3) struktūrą, reiškiančią, jog jų aukštis ir plotis tapo po 244 pikslelius, sudarytus iš 3 spalvų (RGB). Šie paveikslėliai tada buvo paversti į masyvą ir iš anksto apdoroti naudojant *preprocess_input* funkciją iš *tensorf low.keras.applications.vgg16* bibliotekos. Jų pavyzdžiai matomi 32 paveikslėlyje.



32 pav. Paveikslėlių iš duomenų rinkinio pavyzdžiai.

Šie 50 paveikslėlių buvo užpulti tiksline ir netiksline PGD atakomis. Originaliems ir užpultiems paveikslėliams buvo skaičiuojamos SHAP vertės naudojant *shap.GradientExplainer* metodą su parametrais:

- paketo dydis (*batch_size*) = 4;
- lokalus išlyginimas (l*ocal_smoothing*) = 0,1.

SHAP *GradientExplainer* metodas yra specialiai sukurtas giliųjų neuroninių tinklų paaiškinimui. Vietoje to, kad būtų skaičiuojamos tikslios Shapley vertės (kas būtų skaičiavimo požiūriu ne-

praktiška didelės dimensijos įvestims), *GradientExplainer* jas aproksimuoja naudodamas modelio gradientus. Viena esminių *GradientExplainer* metodo dalių yra fono duomenų (angl. *background*) rinkinio naudojimas. Fono parametras pateikia įvesties duomenų, pagal kuriuos apskaičiuojami požymių priskyrimai, etaloninį pasiskirstymą (angl. *reference distribution*). Teorinėje formuluotėje skaičiuojant SHAP reikšmes kiekvieno paveikslėlio bruožo reikšmė matuojama lyginant modelio išvestį visose įmanomose kombinacijose įtraukiant arba išmetant nagrinėjamą bruožą, tačiau praktikoje modelis turi turėti apibrėžtą bruožo nebuvimo sąvoką. Tai pasiekiama pakeičiant bruožų reikšmes pavyzdžiais iš fono duomenų rinkinio. Neturint tinkamo fono, gauti paaiškinimai gali būti nestabilūs arba klaidinantys, nes modelis neturi prasmingo požymių nebuvimo ar jų neutralumo supratimo.

Šiame tyrime naudojami foniniai duomenys buvo sukurti sujungiant pakeisto dydžio $(244, 244, 3)$ įvestinį vaizdą ir sintetinius, triukšmu pagrįstus vaizdus. Dešimt sintetinių vaizdų buvo sugeneruoti imant pavyzdžius iš normaliojo pasiskirstymo su vidurkiu 0,5 ir standartiniu nuokrypiu 0,25, kad būtų imituojamas standartinis normalizuotų pikselių reikšmių intervalas. Po sujungimo visi fono duomenys buvo apdoroti naudojant standartinę VGG16 *preprocess_input* funkciją. Įtraukus realų, pakeisto dydžio vaizdą, fonas geriau atspindi realias domeno savybes, o pridėjus atsitiktinio triukšmo pavyzdžius, sukuriama variacija, apsauganti nuo paaiškinimų pritaikymo vienam konkrečiam pavyzdžiui. Jei fono sudarymui būtų naudojami tik paveikslėliai iš turimo rinkinio, būtų gaunami itin tikslūs, bet galimai riboti paaiškinimai. Priešingai, naudojant tik sintetinius duomenis, būtų prarasta realių duomenų atspindimoji galia, todėl paaiškinimai galėtų būti klaidinantys. Sujungus abu metodus, sukuriamas fonas, kuris suderina specifiškumą ir įvairovę.

Skaičiuojant SHAP vertes su *GradientExplainer* metodu, gaunama išvestis yra (1, aukštis, plotis, 3, 1) struktūros. Pirmasis elementas reiškia, kad aiškinamas vienas paveikslėlis, antras ir trečias struktūros elementai nurodo paveikslėlio aukštį ir plotį, ketvirtasis elementas paaiškina, kad kiekvienam individualiam piksleliui yra priskiriamos trys SHAP reikšmės kiekvienai RGB spalvai, o penktasis elementas nurodo, kad aiškinama vienos klasės prognozė. Norint gauti intuityvesnes ir lengviau interpretuojamas SHAP reikšmes, kiekvieno pikselio trys reikšmės yra sujungiamos į vieną pikslelio svarbos vertę. Ši agregacija atliekama dviem etapais. Pirmiausia, absoliučios SHAP reikšmės sumuojamos per RGB spalvas. Tai užtikrina, kad teigiami ir neigiami indėliai, kurie galėtų tarpusavyje panaikinti vienas kitą, būtų tinkamai įvertinti pagal jų dydį. Antrame žingsnyje gautoms reikšmėms yra skaičiuojamas vidurkis likusioje kanalo dimensijoje. Šių skaičiavimų išvestis yra dviejų dimensijų matrica, kurioje kiekvienas elementas atitinka tam tikrą įvesties vaizdo pikselį, o jo vertė žymi vidutinį to pikselio indėlio dydį per visus RGB spalvų komponentus.

4.2.2. Skaitinės metrikos

Gautoms pikselių svarbos vertėms buvo apskaičiuotos šios metrikos:

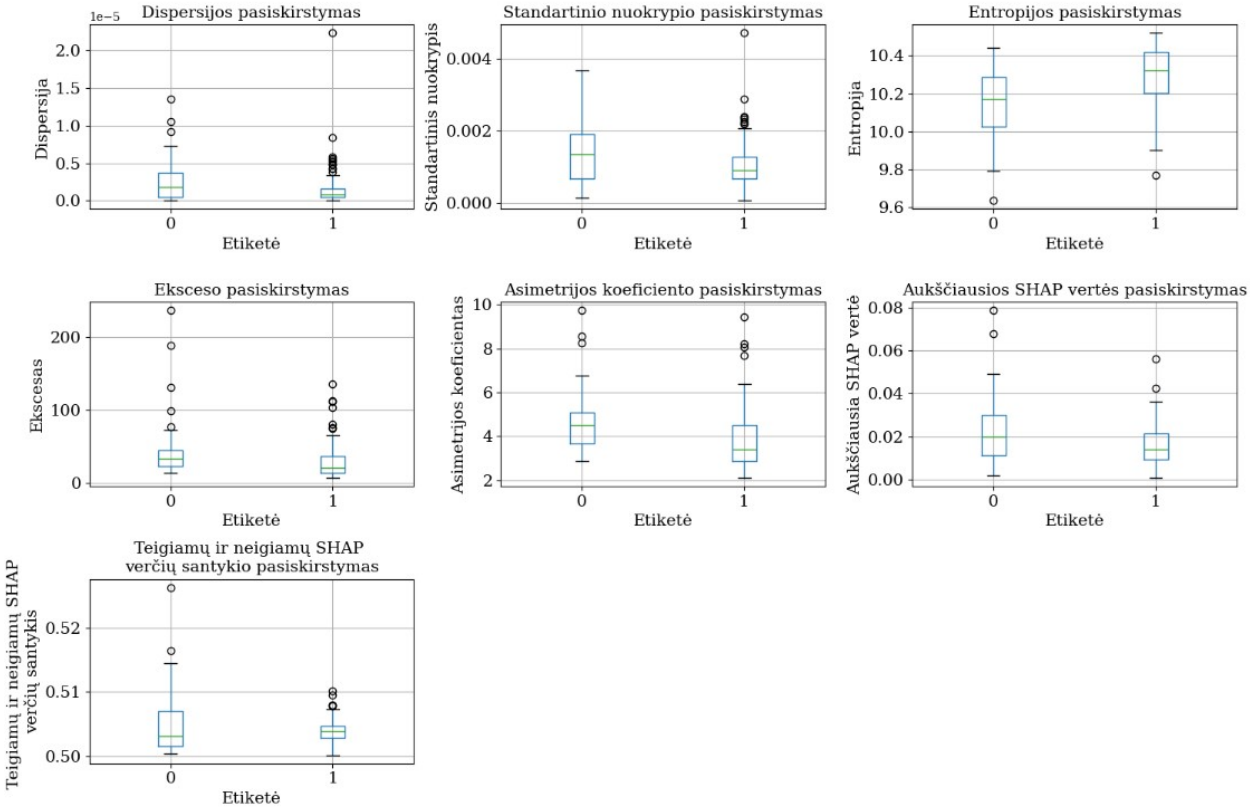
- **Dispersija** (angl. *variance*). Šis matas parodo kaip duomenys pasiskirsto aplink vidurkį. Tyrimo metu yra skaičiuojama, kiek kiekvieno pikselio reikšmingumo vertė yra nutolusi nuo visų pikselių reikšmingumo verčių vidurkio. Pirmiausia, skaičiuojamas visų reikšmių vidurkis, o tada sumuojami kvadratu pakelti skirtumai tarp kiekvieno pikselio reikšmės ir gauto

vidurkio. Galiausiai suma yra padalijama iš pikselių skaičiaus. Didesnė dispersija rodo, kad pikselių reikšmės yra labiau išsklaidytos, daugiau paveikslėlio požymių turi mažesnę įtaką modelio sprendimui, o mažesnė dispersija rodo, kad pikseliai yra panašesni, mažesni paveikslėlio bruožai turi didesnę reikšmę paveikslėlio paaiškinimui.

- **Standartinis nuokrypis** (angl. *standard deviation*). Šis matas taip pat padeda įvertinti, kaip duomenys pasiskirsto aplink vidurkį. Standartinis nuokrypis iš esmės yra reikšmė, gaunama iš dispersijos ištraukus kvadratinę šaknį. Didesnis standartinis nuokrypis taip pat parodo, kad pikselių reikšmės yra labiau išsklaidytos, o mažesnis standartinis nuokrypis patvirtina, kad pikselių reikšmės paaiškinimui yra panašesnės.
- **Entropija** (angl. *entropy*). Šis matas parodo vidutinių pikselių reikšmingumo verčių pasiskirstymo netolygumą arba atsitiktinumą. Tyrimo metu yra skaičiuojama, kaip pikselių reikšmingumo vertės pasiskirsto visame masyve. Pirmiausia kiekvienai reikšmei nustatoma jos atsiradimo tikimybė, o tada apskaičiuojama visų šių tikimybių informacinė reikšmė. Entropija apibrėžiama kaip neigiama visų tikimybių sandaugų su jų logaritmais suma. Didesnė entropija rodo, kad modelio dėmesys yra paskirstytas plačiau per daug pikselių, o mažesnė entropija rodo, kad modelio prognozę lemia keli dominuojantys pikseliai su stipriu reikšmingumu.
- **Ekscesas** (angl. *kurtosis*). Šis matas parodo, kaip stipriai duomenys koncentruojasi aplink vidurkį ir kiek dažnai pasitaiko itin nutolusių reikšmių. Tyrimo metu skaičiuojama, kiek kiekvieno pikselio reikšmingumo vertė, pakelta ketvirtuoju laipsniu, nutolsta nuo bendro visų pikselių reikšmių vidurkio. Pirmiausia apskaičiuojamas vidurkis, tada įvertinamas kiekvieno pikselio reikšmės skirtumas nuo vidurkio ir šis skirtumas pakeliamas ketvirtuoju laipsniu. Gautos reikšmės yra sumuojamos, normalizuojamos pagal dispersiją ir koreguojamos taip, kad normalusis pasiskirstymas turėtų nulį. Didelė eksceso reikšmė rodo, kad nedidelis skaičius pikselių turi labai dideles SHAP reikšmes, o dauguma kitų pikselių turi labai mažas arba beveik nulines vertes. Maža eksceso reikšmė reiškia, kad SHAP reikšmės yra tolygiau pasiskirsčiusios. Tai rodo, kad daugelis savybių panašiai prisideda prie prognozės, bet nė viena nėra labai stipri.
- **Asimetrijos koeficientas** (angl. *skewness*). Šis matas parodo duomenų pasiskirstymo asimetriją aplink vidurkį. Tyrimo metu yra skaičiuojama, kaip pikselių reikšmingumo vertės yra pasislinkusios į vieną ar kitą pusę nuo bendro visų reikšmių vidurkio. Pirmiausia apskaičiuojamas visų reikšmių vidurkis, tada įvertinamas kiekvieno pikselio reikšmės skirtumas nuo vidurkio, pakeliamas trečiuoju laipsniu, ir normalizuojamas pagal standartinį nuokrypį. Teigiama asimetrija rodo, kad daugiau pikselių reikšmių yra žemesnės nei vidurkis, o kelios didelės reikšmės kreipia pasiskirstymą į dešinę. Neigiama asimetrija reiškia, kad dauguma reikšmių yra aukštesnės už vidurkį, o keletas mažų reikšmių kreipia pasiskirstymą į kairę.
- **Trys didžiausios SHAP vertės paveikslėlyje.** Tyrimo metu iš pikselių svarbos verčių sąrašo buvo pasirenkamos trys didžiausią reikšmę turinčios vertės.
- **Santykis tarp paveikslėlio teigiamų ir neigiamų SHAP verčių.** Tyrimo metu skaičiuojamas santykis, apibrėžiantis teigiamų verčių sumos vertę dalinant iš visų absoliučių verčių

sumos reikšmės.

33 paveikslėlyje yra atvaizduotas šių metrikų pasiskirstymas neužpultiems (etiketė 0) ir triukšmu paveiktiems paveikslėliams (etiketė 1) sudarytame duomenų rinkinyje. Nagrinėjant dispersijos pasiskirstymo diagramą yra pastebima, jog abiejų klasių dispersijos medianos yra gana panašios. Triukšmu paveiktų paveikslėlių dispersija šiek tiek mažesnė, o originalūs paveikslėliai turi daugiau išskirčių – tai gali reikšti, kad triukšmas suvienodina SHAP reikšmes, mažindamas jų išsibarstymą. Standartinio nuokrypio diagramoje matome panašius rezultatus – triukšmu paveikti paveikslėliai turi mažesnį standartinį nuokrypį. Entropijos pasiskirstymo diagramoje matome, jog triukšmu paveiktų paveikslėlių entropija yra aukštesnė, kas rodo, kad SHAP reikšmės pasiskirsto tolygiau ir sumažėja labai aiškiai išsiskiriančių pikselių, nes triukšmas išsklaido reikšmių svarbą. Eksceso diagramoje galima įžvelgti eksceso svyravimų ir daug išskirčių. Šiuo atveju sklaidos skirtumas tarp originalių ir triukšmu paveiktų paveikslėlių nėra akivaizdus, tačiau skirstinys triukšmu paveiktuose paveikslėliuose yra mažesnis. Asimetrijos koeficiento pasiskirstymo diagramoje abi klasės rodo didelę teigiamą asimetriją, bet triukšmu paveiktuose paveikslėliuose asimetrijos koeficiento skirstinys yra mažesnis, o sklaida panašaus dydžio. Tai reiškia, kad triukšmas sumažina reikšmių pasiskirstymo pasvirimą į vieną pusę. Lyginant aukščiausių SHAP verčių pasiskirstymą, galima pastebėti, jog originalūs paveikslėliai turi aukštesnes maksimalias SHAP vertes nei triukšmu paveikti, o nagrinėjant teigiamų ir neigiamų verčių santykio pasiskirstymą matyti, kad neužpultų paveikslėlių sklaida yra ženkliai didesnė nei triukšmu paveiktų.



33 pav. Skaitinių metrikų pasiskirstymai.

1 lentelėje yra pateikti kiekvienos iš metrikų pasiskirstymo vidurkiai ir medianos. Triukšmo nepaveiktiems paveikslėliams būdingos didesnės dispersijos (2,5849e-06) ir standartinio nuokrypio

(0,0014) vidurkių reikšmės lyginant su triukšmu paveiktais paveikslėliais (1,6253e-06 ir 0,0011). Tai rodo, kad triukšmo poveikis lemia labiau išsklaidytą SHAP reikšmių pasiskirstymą, iš ko galima spręsti, kad esant įvesties trikdžiams sumažėja požymių svarba paaiškinimui. Entropijos, kuri apibūdina pasiskirstymo neapibrėžtumą ar atsitiktinumą, vidurkis padidėjo nuo 10,1547 triukšmu nepaveiktiems paveikslėliams iki 10,2975 triukšmu paveiktiems paveikslėliams. Šis padidėjimas rodo atsitiktinumo padidėjimą esant triukšmui. Ekscesas sumažėja esant triukšmui nuo 44,5591 (nepaveiktų vaizdų vidurkis) iki 28,5348 (triukšmu paveiktų vaizdų vidurkis). Tai rodo, kad esant triukšmui paveisklėlio ekstremalių SHAP verčių pasireiškimas mažėja. Taip pat, asimetrijos koeficiento pasiskirstymo vidurkis sumažėja nuo 4,6765 iki 3,8376. Medianų metrikos rodo panašias tendencijas. Triukšmu nepaveiktų duomenų atveju dispersija, standartinis nuokrypis, ekscesas ir asimetrija yra šiek tiek didesni nei triukšmo paveiktų duomenų atveju, o entropija yra šiek tiek didesnė užpultuose paveikslėliuose. Tai patvirtina tendenciją link labiau simetriško ir mažiau ekstremalaus pasiskirstymo esant triukšmui. Trys aukščiausios paveikslėlio SHAP vertės taip pat sumažėja triukšmo sąlygomis. Pavyzdžiui, vidutinė Shap1 reikšmė sumažėja nuo 0,0228 iki 0,0161 esant triukšmui. Santykio metrika išlieka gana stabili abiem atvejais, rodydama, kad nepaisant triukšmo, teigiamų ir neigiamų SHAP verčių santykis išlieka pastovus.

Apibendrinant, šie rezultatai rodo, kad triukšmas ne tik paveikia modelio prognozes, bet ir sistemingai keičia požymių svarbos pasiskirstymą. Pastebėta, kad triukšmas sukelia didesnį SHAP verčių pasiskirstymo simetriškumą. Tai leidžia daryti prielaidą, kad modelis tampa mažiau jautrus įvesties požymiams esant triukšmui, o tai gali reikšti aiškinimų tikslumo sumažėjimą triukšmingomis sąlygomis.

1 lentelė. Skaitinių metrikų pasiskirstymo vidurkiai ir medianos.

		Dispersija	Standartinis nuokrypis	Entropija	Ekscesas	Asimetrijos koeficientas
Pasiskirstymo vidurkis	Triukšmu nepaveiktas	2,5849e-06	0,0014	10,1547	44,5591	4,6765
	Triukšmu paveiktas	1,6253e-06	0,0011	10,2975	28,5348	3,8376
Pasiskirstymo mediana	Triukšmu nepaveiktas	1,6253e-06	0,0013	10,1703	33,4352	4,5123
	Triukšmu paveiktas	8,2467e-07	0,0009	10,3207	20,5101	3,4191

		Shap1	Shap2	Shap3	Santykis
Pasiskirstymo vidurkis	Triukšmu nepaveiktas	0,0228	0,0249	0,0294	0,5050
	Triukšmu paveiktas	0,0161	0,0171	0,0189	0,5040
Pasiskirstymo mediana	Triukšmu nepaveiktas	0,0199	0,0211	0,0225	0,5031
	Triukšmu paveiktas	0,0141	0,0147	0,0163	0,5038

Šioms metrikoms taip pat buvo paskaičiuotos t-statistikos ir p-reikšmės naudojant *scipy.stats.ttest_ind* funkciją siekiant įvertinti, ar tarp originalių ir triukšmu paveiktų paveikslėlių kiekvienos metrikos atžvilgiu egzistuoja statistiškai reikšmingas skirtumas. 2 lentelėje yra pateiktos t-statistikos ir p-reikšmės kiekvienai iš metrikų. Gauti rezultatai patvirtina, kad triukšmo perturbacijos sukelia statistiškai reikšmingus SHAP reikšmių pokyčius paveikslėliuose. Dispersijos, standartinio nuokrypio, entropijos, eksceso, asimetrijos bei trijų aukščiausių paveikslėlio SHAP verčių p-reikšmės yra žymiai mažesnės už įprastą statistinio reikšmingumo ribą ($p < 0,05$), tai rodo, jog visos šitos metrikos reikšmingai skiriasi triukšmu nepaveiktiems ir triukšmu paveiktiems paveikslėliams. Ypač maža entropijos p-reikšmė nurodo, kad entropijos metrikos pakitimas tarp neužpultų ir užpultų paveikslėlių duomenų yra didelis ir labai reikšmingas. Teigiamų ir neigiamų SHAP verčių santykis išlieka stabilus, jo p-reikšmė nėra mažesnė už 0,05 (0,0963), vadinasi statistiškai reikšmingo skirtumo nebuvo pastebėta.

2 lentelė. Skaitinių metrikų t-statistikos ir p-reikšmės rezultatai.

	Dispersija	Standartinis nuokrypis	Entropija	Ekscesas	Asimetrijos koeficientas
t-statistika	2,0497	2,1890	-5,0268	2,9773	3,3839
p-reikšmė	0,04216	0,0302	1,4214e-06	0,0034	0,0009

	Shap1	Shap2	Shap3	Santykis
t-statistika	3,2980	3,5339	3,4456	1.6736
p-reikšmė	0,0012	0,0005	0,0007	0.0963

Iš paveikslėliams apskaičiuotų metrikų sudaryta *pandas* bibliotekos objektas *pd.DataFrame* (dvimatė duomenų struktūra) ir kiekvienai eilutei priskirtos etiketės 0 (originalių paveikslėlių duomenims – 50 eilučių) ir 1 (atakos pakeistų paveikslėlių duomenims – 100 eilučių (50 tikslinės atakos ir 50 netikslinės atakos duomenų)). Šią duomenų struktūrą sudaro 150 eilučių ir 10 stulpelių (34 pav.).

	dispersija	standartinis_nuokrypis	entropija	ekscesas	asimetrijos_koef	shap1	shap2	shap3	santykis	etiketė
0	4.407496e-06	0.002099	10.370796	14.090495	2.978640	0.024596	0.025048	0.027135	0.501563	0
1	3.582093e-07	0.000599	10.198949	27.029932	3.888997	0.010712	0.011000	0.011376	0.510539	0
2	1.535411e-06	0.001239	9.635054	188.296759	9.750258	0.032077	0.041241	0.053396	0.502326	0
3	2.680576e-06	0.001637	10.084447	40.319498	4.922755	0.028566	0.029342	0.032008	0.501052	0
4	1.817940e-06	0.001348	10.290700	37.314588	4.689716	0.023063	0.024757	0.025645	0.503198	0
...	...	...	...	...	...	...	...	...	...	...
145	7.053510e-07	0.000840	10.202955	74.373910	6.205541	0.020464	0.021536	0.023468	0.504731	1
146	8.218891e-07	0.000907	10.298350	11.193302	2.753237	0.010253	0.010867	0.012780	0.504376	1
147	8.171736e-07	0.000904	10.277766	35.668446	4.735602	0.014985	0.015307	0.015705	0.504299	1
148	5.962370e-07	0.000772	10.380454	23.912562	3.732538	0.011461	0.012834	0.013736	0.504119	1
149	8.556285e-07	0.000925	10.314126	23.488073	3.677730	0.014142	0.014754	0.018968	0.503520	1

34 pav. Duomenų objekto *pd.DataFrame* struktūra.

4.2.3. Klasifikacija

Tyrimo metu pasitelkiant iš metrikų sudarytą duomenų objektą buvo siekiama apmokyti naują modelį suklasifikuoti pateiktą paveikslėlį į vieną iš dviejų klasių ir taip įvertinti, ar paveikslėlis yra paveiktas triukšmu. Paveikslėlių klasifikavimui pasirinkti du mašininio mokymosi modeliai skirti duomenų klasifikavimui: *RandomForestClassifier* klasifikatorius iš *sklearn* bibliotekos ir Yandex kompanijos sukurtas atvirojo kodo *CatBoostClassifier* klasifikatorius. Šių klasifikatorių pasiektiems rezultatams įvertinti buvo pasirinktos šios metrikos:

- **Tikslumas** (angl. *accuracy*). Šis rodiklis apibrėžiamas kaip teisingai klasifikuotų pavyzdžių dalis visų pavyzdžių atžvilgiu ir naudojamas kaip pagrindinis modelio tikslumo matas. Geras tikslumo rodiklis priklauso nuo sprendžiamos problemos – subalansuotuose duomenų rinkiniuose tikslumas viršijantis 90 % paprastai laikomas puikiu, o sudėtingesnėse ar nesubalansuotose užduotyse net 70–80 % tikslumas gali reikšti stiprius rezultatus. Vis dėlto, vien

tikslumas gali būti klaidinantis esant nesubalansuotom klasėm, todėl tikslumas šio tyrimo metu vertinamas kartu su kitais išsamesniais rodikliais [GBC+16].

- **Precizija** (angl. *precision*). Tai metrika, kuri matuoja modelio teisingai klasifikuotų teigiamų atvejų dalį iš visų atvejų, kurie buvo klasifikuoti kaip teigiami. Triukšmo pagrindu atliktų atakų aptikimo kontekste precizija parodo, kaip tiksliai klasifikatorius klasifikuoja vaizdą kaip užpultą. Aukšta precizija rodo, kad klasifikatorius retai klaidingai klasifikuoja švarius vaizdus kaip užpultus.
- **Atpažinimas** (angl. *recall*). Šis rodiklis nurodo, kaip gerai klasifikatorius sugeba identifikuoti visus tikruosius teigiamus atvejus. Jis matuoja tikrų užpultų vaizdų dalį, kurią klasifikatorius sėkmingai aptinka. Aukštas atpažinimas yra ypač svarbus scenarijuose, kai reikia nustatyti visus triukšmo pagrindu atliktus užpuolimus, net jei tai reiškia, kad keli švarūs paveikslėliai bus klasifikuoti kaip paveikti triukšmo.
- **F-1 rodiklis**. Jis nusako tikslumo ir atpažinimo pusiausvyrą viena metrika, kuri atspindi tiek klasifikatoriaus gebėjimą teisingai identifikuoti teigiamus atvejus, tiek jo gebėjimą aptikti visus tikruosius teigiamus atvejus. Ši metrika ypač naudinga, kai duomenų rinkinys nėra gerai subalansuotas, nes ji užtikrina, kad nei klaidingi teigiami, nei klaidingi neigiami atvejai nedarys didelės įtakos modelio veikimui.
- **ROC kreivė** (angl. *Receiver Operating Characteristic*). Tai yra teisingų teigiamų spėjimų dažnio (TPR, angl. *True Positive Rate*) ir klaidingų teigiamų spėjimų dažnio (angl. *False Positive Rate*, FPR) grafikas nustatytose slenksčio vertėse. Modelis su gera ROC kreive sugeba atskirti užpultus ir neužpultus vaizdus su minimaliu klaidingų teigiamų ir klaidingų neigiamų atvejų skaičiumi. Kuo arčiau ROC kreivė yra prie viršutinio kairiojo grafiko kampo, tuo didesnis klasifikatoriaus tikslumas.
- **AUC balas** (angl. *Area Under the Curve*). Balas, arba plotas po kreive, kiekybiškai įvertina bendrą klasifikatoriaus veikimą, matuodamas plotą po ROC kreive. Šis balas apibendrina, kaip gerai klasifikatorius sugeba atskirti dvi klases (užpultus ir neužpultus vaizdus) be jokių specifinių slenksčių. Aukštas AUC balas rodo, kad klasifikatorius efektyviai atskiria dvi klases per visus galimus slenksčius. Ideali ROC kreivė yra ta, kurios AUC vertė yra 1.

Šios metrikos buvo pasirinktos, nes jos suteikia išsamų klasifikatoriaus veikimo įvertinimą, padedantį nustatyti jo gebėjimą aptikti triukšmo atakas paveikslėliuose.

4.2.3.1. *RandomForest* veikimo principas ir rezultatai

RandomForest klasifikatorius [PKP18] veikia kuriant kelis sprendimo medžius (angl. *decision tree*) mokymo metu, kiekvieną apmokant su atsitiktiniu duomenų pogrupiu. Kiekvienas sprendimų medis priima sprendimą, į kurią iš dviejų klasių (0 ar 1) patenka vaizdas. Klasifikuojant naują vaizdą, *RandomForest* sujungia visų atskirų medžių prognozes ir pagal daugumą balsų priskiria paveikslėliui klasę. Kelių medžių kombinacija padidina modelio prognozių tikslumą ir sumažina persimokymo riziką. Šis klasifikatorius buvo pasirinktas tyrimui dėl savo patikimumo ir populiarumo.

Šio tyrimo metu, duomenų rinkinys į mokymo ir testavimo duomenis buvo išskaidytas nau-

dojant funkciją *train_test_split* iš *sklearn.model_selection* modulio su parametrais:

- Testavimo duomenų dydis (*test_size*) = 0,2. Šis parametras nusako, kokiu santykiu bus išskirstyti duomenys į mokymo ir testavimo duomenų aibes. Šiam tyrimui pasirinkta 80 % duomenų naudoti modelio mokymui, o likusius 20 % testavimui.
- Atsitiktinė būsena (*random_state*) = 42. Šis parametras užtikrina, kad duomenys būtų padalinti atkuriamu būdu. Naudojant tą pačią parametro vertę yra gaunamas tas pats duomenų padalinimas.

RandomForest modeliui pateikti požymiai atitinka apskaičiuotas SHAP verčių skaitines metrikas (dispersija, standartinis nuokrypis, entropija, ekscesas, asimetrijos koeficientas, aukščiausia SHAP vertė, antra aukščiausia SHAP vertė, trečia aukščiausia SHAP vertė, teigiamų ir neigiamų SHAP verčių santykis), o duomenų etiketės 0 ir 1, atitinkamai nurodo, ar paveikslėlis yra paveiktas triukšmu.

3 lentelėje pavaizduotoje klasifikacijos matricoje matoma, jog *RandomForest* modelis teisingai atpažino 7 paveikslėlius kaip nepaveiktus triukšmo, tačiau 3 originalius paveisklėlius suklasifikavo kaip paveiktus triukšmo. 1 triukšmu paveiktas paveikslėlis buvo neteisingai atpažintas kaip triukšmu nepaveiktas, o kiti 19 atakuotų paveikslėlių buvo suklasifikuoti teisingai. Ši asimetrija rodo, kad modelis yra atsargus ir gerai sugauna atakas, tačiau kartais klaidingai pažymi originalius vaizdus kaip atakuotus.

3 lentelė. *RandomForest* klasifikavimo matrica.

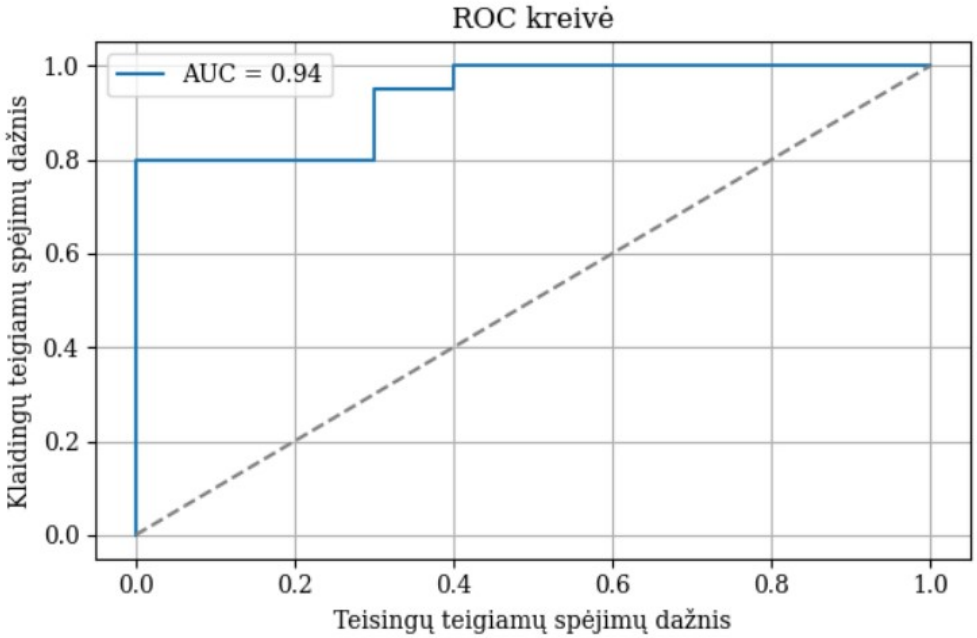
		Tikroji klasė	
		Triukšmu nepaveiktas	Triukšmu paveiktas
Spėta klasė	Triukšmu nepaveiktas	7	3
	Triukšmu paveiktas	1	19

Iš klasifikacijos matricos rezultatų buvo apskaičiuoti *RandomForest* modelio tikslumo, precizijos, atpažinimo ir F-1 rodikliai (4 lentelė). Klasifikuodamas triukšmo nepaveiktus paveikslėlius klasifikatorius pasiekė 88 % preciziją, o tai rodo, kad dauguma vaizdų iš tiesų buvo teisingai suklasifikuoti. Vis dėl to, šios klasės atpažinimo rodiklis siekė tik 70 % (30 % triukšmu nepaveiktų vaizdų buvo klaidingai priskirti prie užpultų), kas rodo, kad nemaža dalis triukšmu neužpultų vaizdų buvo neteisingai suklasifikuoti. F-1 rodiklis triukšmo nepaveiktiems vaizdams siekė 0,78, parodydamas šiek tiek nesubalansuotą precizijos ir atpažinimo derinį. Triukšmo paveiktiems vaizdams klasifikatorius išlaikė 86 % preciziją, šiek tiek mažesnę nei triukšmu nepaveiktų vaizdų atveju, tačiau atpažinimo rodiklis siekė net 95 %, parodydamas didelį jautrumą aptinkant atakuotus vaizdus. Šios klasės F-1 rodiklis buvo 0,90, rodantis tvirtą ir subalansuotą klasifikavimo kokybę. Bendras modelio klasifikavimo tikslumas siekė 87 %, kas parodo patikimą klasifikatoriaus veikimą skiriant triukšmo nepaveiktus ir triukšmo paveiktus paveikslėlius. Šie rezultatai pabrėžia modelio polinkį teikti pirmenybę triukšmu užpultų vaizdų aptikimui.

4 lentelė. *RandomForest* klasifikatoriaus rezultatai.

	Precizija	Atpažinimo rodiklis	F-1 rodiklis	Tikslumas
Triukšmu nepaveiktas	0,88	0,70	0,78	0,87
Triukšmu paveiktas	0,86	0,95	0,90	

RandomForest klasifikatoriaus ROC kreivė (35 pav.) rodo stiprų klasifikavimą, AUC reikšmei siekiant 0,94. Tai reiškia, kad modelis pasiekia aukštą teisingų teigiamų spėjimų dažnį, išlaikydamas žemą klaidingų teigiamų spėjimų dažnį įvairiuose klasifikavimo slenksčiuose. Kreivė aiškiai išsidėsčiusi virš įstrižainės, siekianti arti viršutinio kairiojo grafiko kampo, kas rodo gerą modelio gebėjimą atskirti triukšmo paveiktų ir nepaveiktų paveikslėlių klases.



35 pav. *RandomForest* klasifikatoriaus ROC ir AUC vertės.

4.2.3.2. *CatBoost* veikimo principas ir rezultatai

CatBoost klasifikatorius [DEG18] remiasi gradiento didinimo (angl. *gradient boosting*) algoritmu – yra nuosekliai kuriama sprendimų medžių serija, kur kiekvienas naujas medis ištaiso ankstesnio medžio padarytas klaidas. Tai didinimo metodas, kai galutinis modelis yra medžių, kurie kartu pagerina prognozes, grupė. Priešingai nei *RandomForest* klasifikatorius, kuris sprendimų medžius kuria paraleliai, *CatBoost* sprendimų medžiai yra kuriami sekoje, vienas po kito. *CatBoost* klasifikatorius turi integruotus metodus, kurie padeda išvengti modelio persimokymo, ypač dirbant su mažais arba triukšmingais duomenų rinkiniais. Šis klasifikatorius buvo pasirinktas tyrimui dėl savo turimos optimizacijos mažiems duomenų rinkiniams. Tyrimo metu duomenų rinkinys į mokymo ir testavimo aibes buvo padalintas naudojant tą pačią funkciją su tais pačiais parametrais kaip

ir *RandomForest* modelyje. *CatBoost* modeliui pateikti tie patys požymiai ir duomenų etiketės.

5 lentelėje pavaizduotoje klasifikacijos matricoje matoma, jog *CatBoost* modelis teisingai suklasifikavo 7 paveikslėlius kaip nepaveiktus triukšmo, o 3 originalius paveikslėlius suklasifikavo kaip paveiktus triukšmo. Visi 20 triukšmu užpulti paveikslėliai buvo suklasifikuoti teisingai.

5 lentelė. *CatBoost* klasifikavimo matrica.

		Tikroji klasė	
		Triukšmu nepaveiktas	Triukšmu paveiktas
Spėta klasė	Triukšmu nepaveiktas	7	3
	Triukšmu paveiktas	0	20

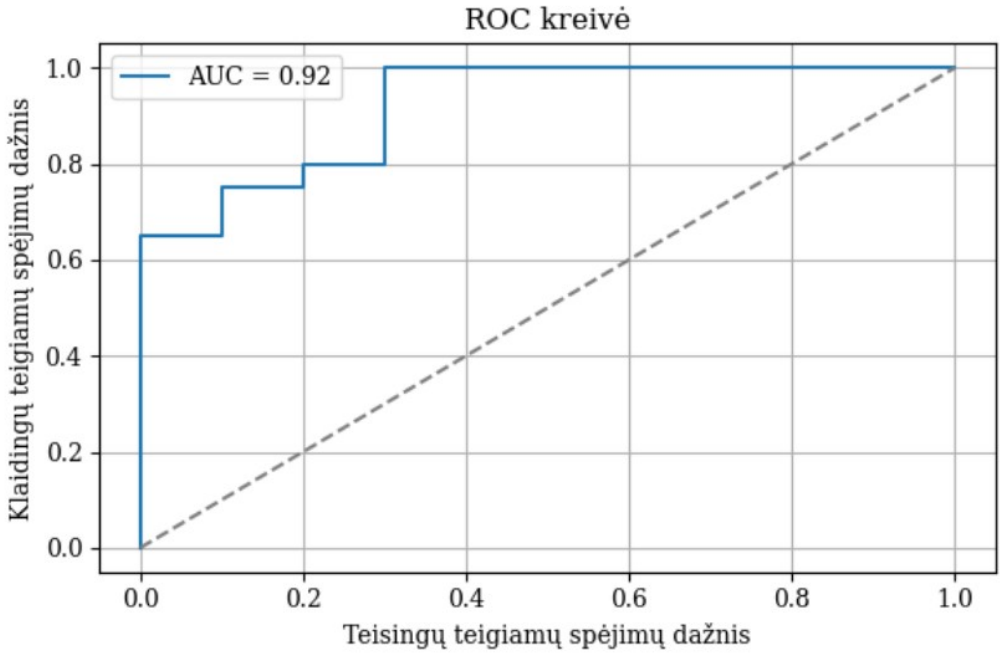
Iš klasifikacijos matricos *CatBoost* modeliui apskaičiuoti tikslumo, precizijos, atpažinimo ir F-1 rodikliai (6 lentelė). Klasifikuodamas triukšmo nepaveiktus paveikslėlius, klasifikatorius pasiekė 100 % preciziją, kas reiškia, kad visi paveikslėliai, kurie buvo suklasifikuoti kaip triukšmo nepaveikti, buvo teisingai priskirti savo klasei. Vis dėlto, kaip ir *RandomForest*, triukšmo nepaveiktų paveikslėlių atpažinimo rodiklis siekė tik 70 %, parodydamas, kad net 30 % šių vaizdų buvo klaidingai priskirti prie triukšmo paveiktų. F-1 rodiklis triukšmo nepaveiktiems vaizdams siekė 0,82, atspindėdamas gerą precizijos ir atpažinimo santykį. Triukšmo paveiktų paveikslėlių klasifikavime klasifikatorius pasiekė 87 % preciziją, šiek tiek žemesnę nei nepaveiktų vaizdų atveju, tačiau atpažinimo rodiklis siekė net 100 %, rodydamas itin aukštą jautrumą aptinkant triukšmu užpultus vaizdus. Šios klasės F-1 rodiklis buvo 0,93, rodantis tvirtą ir subalansuotą klasifikavimo kokybę. Bendras modelio klasifikavimo tikslumas buvo 90 %, kas parodo patikimą klasifikatoriaus veikimą skiriant triukšmo nepaveiktus ir triukšmo paveiktus paveikslėlius. Rezultatai rodo, kad modelis linkęs labai jautriai aptikti triukšmo paveiktus vaizdus, kartu išlaikydamas aukštą preciziją nepaveiktų vaizdų atpažinime.

6 lentelė. *CatBoost* klasifikatoriaus rezultatai.

	Precizija	Atpažinimo rodiklis	F-1 rodiklis	Tikslumas
Triukšmu nepaveiktas	1,00	0,70	0,82	0,90
Triukšmu paveiktas	0,87	1,00	0,93	

CatBoost modelio ROC kreivė (36 pav.) rodo labai gerą klasifikavimo veikimą, AUC reikšmei siekiant 0,92. Nors ši reikšmė šiek tiek mažesnė nei *RandomForest* modelio, ji vis tiek rodo aukštą modelio tikslumą skiriant abi paveikslėlių kategorijas. Kreivė išlieka gerokai aukščiau už

įstrižainę, kas pabrėžia stiprų jautrumą ir specifiškumą. Nepaisant nedidelių nukrypimų nuo idealaus klasifikatoriaus, *CatBoost* modelis demonstruoja tvirtus prognozavimo gebėjimus ir gali būti laikomas stipriu kandidatu triukšmo aptikimo užduotims.



36 pav. *CatBoost* klasifikatoriaus ROC ir AUC vertės.

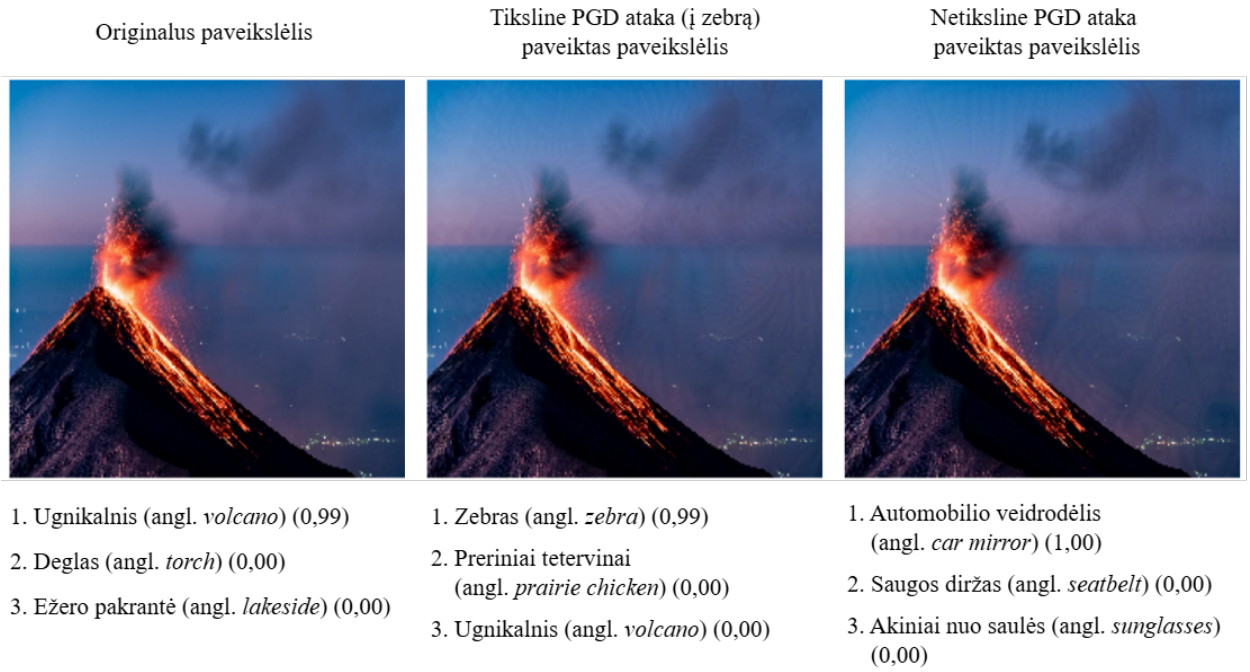
4.2.3.3. Modelių palyginimas

Šiame tyrime buvo vertinamas *RandomForest* ir *CatBoost* klasifikatorių našumas klasifikuojant triukšmo nepaveiktus ir paveiktus paveikslėlius. Rezultatai parodė, kad abu klasifikatoriai pasiekė aukštus rodiklius pagal pasirinktus vertinimo kriterijus, tačiau buvo pastebėti tam tikri skirtumai. *RandomForest* pasiekė AUC reikšmę 0,94, šiek tiek aplenkdamas *CatBoost*, kurio AUC siekė 0,92. *RandomForest* pasiekė 88 % preciziją triukšmu nepaveiktiems ir 86 % paveiktiems paveikslėliams. Tuo tarpu *CatBoost* pasižymėjo puikia precizija (100 %) triukšmo nepaveikties vaizdams, bet šiek tiek mažesne (87 %) užpultiems paveikslėliams. Pagal atpažinimo rodiklį, *CatBoost* reikšmingai aplenkė *RandomForest*, pasiekdamas puikų 100 % atpažinimą triukšmo paveiktiems paveikslėliams, kai *RandomForest* pasiekė 95 %. Neužpultiems paveikslėliams abu modeliai pasiekė tą pačią reikšmę 70 %. Lyginant F-1 rodiklio reikšmes, *CatBoost* vėl demonstravo geresnį rezultatą – 0,82 (triukšmu nepaveiktiems paveikslėliams) ir 0,93 (užpultiems paveikslėliams), palyginti su *RandomForest* 0,78 (neužpultiems) ir 0,90 (užpultiems). Vertinant bendrą modelių klasifikavimo tikslumą, *CatBoost* šiek tiek aplenkė *RandomForest* – atitinkamai 90 % ir 87 %.

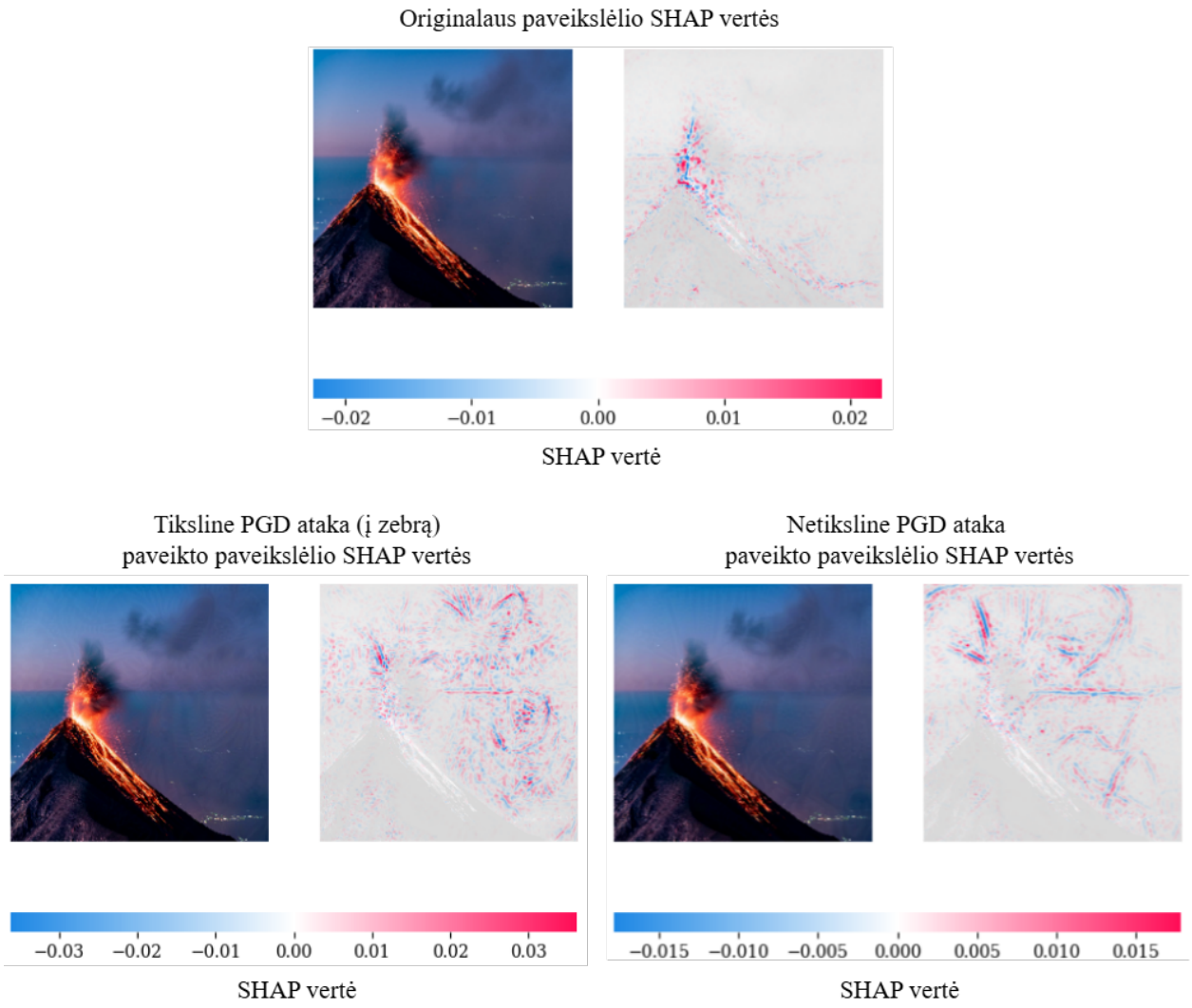
Šie rezultatai rodo, kad abu modeliai yra veiksmingi, tačiau *CatBoost* pasižymi šiek tiek geresne pusiausvyra tarp precizijos, atpažinimo rodiklio ir F-1 rodiklio, ypač kai paveikslėlis yra paveiktas triukšmo ataka. Visgi, *RandomForest* turi nedidelį pranašumą pagal AUC, rodantį šiek tiek geresnį bendrą klasių išrikiavimo gebėjimą. Todėl modelio pasirinkimas gali priklausyti nuo konkrečių taikymo prioritetų – ar labiau pageidaujamas aukštesnis AUC, ar geresnė precizijos ir atpažinimo pusiausvyra.

4.2.3.4. Testavimo rezultatai

Apmokius abu mašininio mokymosi modelius, jų veikimo rezultatai buvo įvertinti su nauju paveikslėliu. Pasirinktam ugnikalnio vaizdui buvo apskaičiuotos jį apibūdinančios metrikos, atitinkančios duomenų rinkinyje turėtas reikšmes. Šis paveikslėlis tada buvo užpultas naudojant tikslinę (į zebrą) ir netikslinę PGD atakas, naujiems triukšmu paveiktiems paveikslėliams taip pat buvo apskaičiuotos reikiamos metrikos. VGG16 neuroninio tinklo klasifikavimo rezultatai originaliam bei abiem triukšmu paveiktiems paveikslėliams yra pavaizduoti 37 paveikslėlyje. Testavimo metu triukšmu paveiktuose paveikslėliuose galima įžiūrėti šiokius tokius iškraipymus fone, tačiau pagrindinis paveikslėlio objektas – ugnikalnis – vis dar yra žmogui puikiai atpažįstamas objektas, vizualiai nepakitęs dėl uždėto triukšmo. Prieš ataką klasifikuotas kaip ugnikalnis su pasitikėjimo reikšme 0,99, objektas po tikslinės atakos su tokiu pačiu pasitikėjimo rodikliu tapo zebru, o įvykdžius netikslinę PGD ataką, pasikeitė į automobilio veidrodėlį. SHAP verčių šilumos žemėlapiai yra pavaizduoti 38 pav. Juose galima pastebėti, kad originalaus paveikslėlio žemėlapyje reikšmingomis vertėmis buvo pažymėti ugnikalnio kontūrai, o triukšmu paveiktuose paveikslėliuose jos yra plačiau pasiskirsčiusios fone, nebėra išskiriama ugnikalnio forma.



37 pav. Tyrime naudotas originalus ugnikalnio paveikslėlis ir tas pats paveikslėlis užpultas tiksline ir netiksline PGD atakomis bei jų klasifikavimo rezultatai.



38 pav. SHAP verčių šilumos žemėlapiai sugeneruoti originaliam (viršutinė eilutė), tiksline PGD ataka užpultam (pirmas paveikslėlis apatinėje eilutėje) ir netiksline PGD ataka užpultam (antras paveikslėlis apatinėje eilutėje) ugnikalnio paveikslėliams.

7 lentelėje yra aprašyti *RandomForest* ir *CatBoost* klasifikatorių rezultatai. *RandomForest* šiuo atveju teisingai suklasifikavo originalų ir tiksline PGD ataka paveiktą paveikslėlį, tačiau teigė, jog netiksline PGD ataka paveiktam paveikslėliui triukšmo uždėta nebuvo. Tuo tarpu *CatBoost* visus tris paveikslėlius suklasifikavo teisingai. Lentelėje taip pat galima matyti ne tik klasifikavimo rezultatus, bet ir gautą tikimybę, jog paveikslėlis yra paveiktas triukšmo. Originaliam paveikslėliui ši tikimybė abiejų modelių atžvilgiu buvo tokia pati, tikslinės PGD atakos paveiktam paveikslėliui labai panaši, o labiausiai skyrėsi netiksline PGD ataka paveiktam paveikslėliui. *RandomForest*, šią tikimybę paskaičiavęs kaip 0,3 nusprendė, kad paveikslėlis yra nepaveiktas, o *CatBoost*, pasiekęs 0,51 priėmė teisingą sprendimą. Nagrinėjant šiuos rezultatus galima spręsti, jog abu triukšmu paveikti paveikslėliai sukėlė iššūkių abiems klasifikavimo modeliams – gautos jų tikimybės nėra labai aukštos.

7 lentelė. Ugnikalnio paveikslėlio klasifikavimo rezultatai.

	Originalus paveikslėlis	Tikslinė PGD ataka	Netikslinė PGD ataka
RandomForest	Triukšmu nepaveiktas P(užpultas) = 0,14	Triukšmu paveiktas P(užpultas) = 0,53	Triukšmu nepaveiktas P(užpultas) = 0,30
CatBoost	Triukšmu nepaveiktas P(užpultas) = 0,14	Triukšmu paveiktas P(užpultas) = 0,55	Triukšmu paveiktas P(užpultas) = 0,51

Vertinant atsitiktinai pasirinkto paveikslėlio rezultatus galima teigti, kad abi triukšmo atakos buvo sėkmingos ir, nors paveikslėliuose yra pastebimas nežymus fono iškraipymas, žmogui nebūtų sunku atpažinti jame pavaizduotą ugnikalnį. Aiškinant juos SHAP metodu yra pastebimi pasikeitę svarbiausių požymių išsidėstymo skirtumai – jie nebėra susitelkę ties ugnikalnio kontūrais ir išsiliejusia lava, bet yra pasklidę už ugnikalnio esančiame fone. Naudojant *RandomForest* klasifikatorių teisingai identifikuojami du paveikslėliai iš trijų, o *CatBoost* klasifikatorius teisingai suklasifikavo visus testavimo metu naudotus paveikslėlius.

REZULTATAI IR IŠVADOS

Atliktų tyrimų metu buvo išbandyti ir vizualiai palyginti trys dirbtinių neuroninių tinklų paaiškinamumo ir interpretuojamumo metodai – LIME, SHAP ir Grad-CAM bei atlikta SHAP metodu gautų pikselių svarbos verčių metrikų ir jų pritaikomumo triukšmo atakos atpažinimui analizė. Tyrimo pradžioje buvo pasirinkti trys skirtingus objektus vaizduojantys paveikslėliai, jie buvo suklasifikuoti naudojant iš anksto su *ImageNet* duomenų rinkiniu apmokytą VGG16 neuroninio tinklo modelį. Gauti rezultatai buvo paaiškinti naudojant šiuos tris metodus. Vėliau tie paveikslėliai buvo paveikti triukšmu naudojant tikslinę ir netikslinę PGD metodo atakas. Tikslinės atakos metu buvo bandoma paveikslėlį paveikti triukšmu taip, kad modelis pradėtų jį klasifikuoti kaip zebrą, o netikslinės PGD atakos metu nebuvo svarbu, kokia neteisinga klasė bus priskirta paveikslėliui. Šiems užpultiems paveikslėliams taip pat buvo pritaikyti tie patys trys paaiškinamumo ir interpretuojamumo metodai ir vykdoma gautų paaiškinimų analizė originaliems ir triukšmu užpultiems paveikslėliams. Analizuojant LIME, SHAP ir Grad-CAM metodų vaizdinius rezultatus, buvo vizualiai įvertinta, kiek atvaizduoti svarbiausi požymiai atitinka objekto, kuris vaizduojamas paveikslėlyje, savybes. Taip pat buvo siekiama pastebėti šablonus ir konkrečius skirtumus tarp paveikslėlių paaiškinimų. Tyrimas toliau buvo vykdomas analizuojant SHAP metodu gautas skaitines pikselių svarbumo vertes su 50 paveikslėlių. Iš pradžių buvo suskaičiuotos kiekvieno iš jų pikselių svarbumo vertės ir pasirinktos metrikos joms. Tada paveikslėliams buvo uždėtas triukšmas naudojant tikslinę ir netikslinę PGD metodo atakas bei gauti skaitiniai rezultatai jiems. Iš triukšmu nepaveiktų ir paveiktų paveikslėlių metrikų buvo sudarytas naujas duomenų rinkinys, kuris buvo naudojamas apmokyti *RandomForest* ir *CatBoost* klasifikatorius. Pateikus klasifikatoriams dar nematytą ugnikalnio paveikslėlį bei tiksliniu ir netiksliniu triukšmu užpultas jo versijas, *RandomForest* klasifikatorius teisingai identifikavo du paveikslėlius iš trijų, o *CatBoost* klasifikatorius teisingai suklasifikavo visas testavimo metu naudotas paveikslėlio versijas.

Analizės metu buvo padarytos šios išvados:

1. LIME, SHAP ir Grad-CAM metodai gali suteikti įžvalgų apie modelio veikimą ir padėti suprasti, kurios paveikslėlio sritys daro įtaką klasifikacijos rezultatams. LIME ir SHAP metodai yra pakankamai detalūs išskiriant požymius ir jų reikšmingumą gautiems rezultatams, o Grad-CAM metodu gauti paaiškinimai yra labiau abstraktūs. Lyginant LIME ir SHAP metodais gautų segmentų, darančių teigiamą arba neigiamą įtaką, reikšmingumą, buvo pastebėta, jog originalių paveikslėlių segmentai gali būti iki kelių šimtų kartų reikšmingesni.
2. Naudojant SHAP paaiškinamumo ir interpretuojamumo metodą yra pastebima, kad originalaus paveikslėlio paaiškinimuose ryškiausi segmentai yra matomi didžiausios tikimybės klasėje, tuo tarpu triukšmu užpultuose paveikslėliuose SHAP vertės yra panašios visuose keturiuose didžiausias tikimybes turinčių klasių paaiškinimuose.
3. Atlikus statistinę analizę patvirtinta, kad dauguma SHAP verčių metrikų reikšmingai pakinta esant triukšmui, dėl to pikselių svarbos vertėms apskaičiuotos metrikos gali būti naudojamos kaip požymiai klasifikatoriui, siekiant nustatyti ar paveikslėlis buvo užpultas triukšmu.
4. Entropija padidėjo, o pikselių svarbos reikšmės, dispersija, ekscesas bei asimetrijos koeficientas sumažėjo paveikslėlyje esant triukšmui. Tai rodo, kad triukšmas mažina požymių

svarbą modelio klasifikacijoje, triukšmu paveiktų paveikslėlių SHAP reikšmės yra pasiskirsčiusios tolygiau.

5. Atlikus vizualią ir skaitinę modelio rezultatų paaiškinimų analizes, abiem atvejais matoma, kad esant triukšmui sugeneruotas paaiškinimas mažiau koncentruojasi į keletą specifinių požymių, yra labiau atsitiktinai pasklidęs visame paveikslėlyje.
6. Apmokytas *RandomForest* klasifikatorius parodė 87 % bendrą tikslumą ir 0,94 AUC reikšmę, *CatBoost* pasiekė 90 % tikslumą ir 0,92 AUC reikšmę. Testavimo metu su nauju vaizdu (ugnikalniu) *CatBoost* tinkamai suklasifikavo visus vaizdus, o *RandomForest* suklydo klasifikuodamas netikslinės atakos atvejį. Tai įrodo, kad pasirinktų metrikų skaičiavimas yra vienas iš būdų atpažinti paveikslėlius, paveiktus triukšmu, tačiau būtų tikslinga toliau ieškoti papildomų reikšmingų SHAP verčių metrikų, kurios leistų padidinti klasifikatorių sprendimų užtikrintumą.

ŠALTINIŲ SĄRAŠAS

[AA22] Tanseem N Abu-Jamie ir Samy S Abu-Naser. Classification of sign-language using vgg16, 2022.

[AAE+23] Sajid Ali, Tamer Abuhmed, Shaker El-Sappagh, Khan Muhammad ir k.t. Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence. *Information Fusion*, 99:101805, 2023. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2023.101805. URL: https://www.sciencedirect.com/science/article/pii/S1566253523001148.

[AAS+22] Sara A Althubiti, Fayadh Alenezi, S Shitharth, Sangeetha K ir Chennareddy Vijay Simha Reddy. Circuit manufacturing defect detection using VGG16 convolutional neural networks. *Wireless Communications and Mobile Computing*, 2022:1–10, 2022.

[Abr05] Ajith Abraham. Artificial neural networks. *Handbook of measuring system design*, 2005.

[AZC+23] Antonio Luca Alfeo, Antonio G. Zippo, Vincenzo Catrambone, Mario G.C.A. Cimino, Nicola Toschi ir Gaetano Valenza. From local counterfactuals to global feature importance: efficient, robust, and model-agnostic explanations for brain connectivity networks. *Computer Methods and Programs in Biomedicine*, 236:107550, 2023. ISSN: 0169-2607. DOI: https://doi.org/10.1016/j.cmpb.2023.107550. URL: https://www.sciencedirect.com/science/article/pii/S0169260723002158.

[BAM+22] Teo Beker, Homa Ansari, Sina Montazeri, Qian Song ir Xiao Xiang Zhu. Explainability analysis of CNN in detection of volcanic deformation signal. *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*, p.p. 4851–4854. IEEE, 2022.

[BGM+20] Niklas Bussmann, Paolo Giudici, Dimitri Marinelli ir Jochen Papenbrock. Explainable AI in fintech risk management. *Frontiers in Artificial Intelligence*, 3:26, 2020.

[BLZ+21] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen ir Qian Wang. Recent Advances in Adversarial Training for Adversarial Robustness, 2021. arXiv: 2102.01356 [cs.LG].

[BSC21] Florentin Bieder, Robin Sandkühler ir Philippe C. Cattin. Comparison of Methods Generalizing Max- and Average-Pooling, 2021. arXiv: 2103.01746 [cs.CV].

[CLG+15] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm ir Noemie Elhadad. Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-Day Readmission. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, p.p. 1721–1730, Sydney, NSW, Australia. Association for Computing Machinery, 2015. ISBN: 9781450336642. DOI: 10.1145/2783258.2788613. URL: https://doi.org/10.1145/2783258.2788613.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li ir Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, p.p. 248–255. Ieee, 2009.

[DEG18] Anna Veronika Dorogush, Vasily Ershov ir Andrey Gulin. CatBoost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*, 2018.

[DSC22] Shiv Ram Dubey, Satish Kumar Singh ir Bidyut Baran Chaudhuri. Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark, 2022. arXiv: 2109.14545 [cs.LG].

[GBC+16] Ian Goodfellow, Yoshua Bengio, Aaron Courville ir Yoshua Bengio. *Deep learning*, tom. 1 numeris 2. MIT press Cambridge, 2016.

[GF17] Bryce Goodman ir Seth Flaxman. European Union Regulations on Algorithmic Decision Making and a "Right to Explanation". *AI Magazine*, 38(3):50–57, 2017-09. ISSN: 2371-9621. DOI: 10.1609/aimag.v38i3.2741. URL: http://dx.doi.org/10.1609/aimag.v38i3.2741.

[GM21] Damien Garreau ir Dina Mardaoui. What does LIME really see in images?, 2021. arXiv: 2102.06307 [cs.LG].

[GSS15] Ian J. Goodfellow, Jonathon Shlens ir Christian Szegedy. Explaining and Harnessing Adversarial Examples, 2015. arXiv: 1412.6572 [stat.ML].

[Gur97] Kevin Gurney. *An introduction to neural networks*. UCL Press, 1997, p.p. 278–280. ISBN: 0-203-45151-1. URL: https://www.inf.ed.ac.uk/teaching/courses/nlu/assets/reading/Gurney_et_al.pdf.

[GWK+18] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma ir k.t. Recent advances in convolutional neural networks. *Pattern recognition*, 77:354–377, 2018.

[HBA+14] Luis Hernández-Callejo, Carlos Baladrón Zorita, Javier Aguiar, Belén Carro, Antonio Sanchez-Esguevillas, Jaime Lloret ir Joaquim Massana. A Survey on Electric Power Demand Forecasting: Future Trends in Smart Grids, Microgrids and Smart Buildings. *Communications Surveys & Tutorials, IEEE*, 16:1460–1495, 2014-04. DOI: 10.1109/SURV.2014.032014.00094.

[HGJ23] John Harshith, Mantej Singh Gill ir Madhan Jothimani. Evaluating the Vulnerabilities in ML systems in terms of adversarial attacks, 2023. arXiv: 2308.12918 [cs.LG].

[Ima14] Imagenet. Large Scale Visual Recognition Challenge 2014 (ILSVRC2014). 2014. URL: https://image-net.org/challenges/LSVRC/2014/results (tikrinta 2024-06-03).

[YND+18] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do ir Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9:611–629, 2018.

[YXF+18] Jiawen Yang, Fengying Xie, Haidi Fan, Zhiguo Jiang ir Jie Liu. Classification for dermoscopy images using convolutional neural networks based on region average pooling. *IEEE Access*, 6:65130–65138, 2018.

[KBK11] Andrej Krenker, Janez Bešter ir Andrej Kos. Introduction to the artificial neural networks. *Artificial Neural Networks: Methodological Advances and Biomedical Applications. InTech*:1–18, 2011.

[KMG20] Zuzanna Klawikowska, Agnieszka Mikołajczyk ir Michał Grochowski. Explainable ai for inspecting adversarial attacks on deep neural networks. *Artificial Intelligence and Soft Computing: 19th International Conference, ICAISC 2020, Zakopane, Poland, October 12-14, 2020, Proceedings, Part I 19*, p.p. 134–146. Springer, 2020.

[KP20] Aravind Krishnaswamy Rangarajan ir Raja Purushothaman. Disease classification in eggplant using pre-trained VGG16 and MSVM. *Scientific reports*, 10(1):2322, 2020.

[LL17] Scott Lundberg ir Su-In Lee. A Unified Approach to Interpreting Model Predictions, 2017. arXiv: 1705.07874 [cs.AI].

[LLY+21] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng ir Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 33(12):6999–7019, 2021.

[Lun18] Scott Lundberg. SHAP documentation. 2018. URL: https://shap.readthedocs.io/en/latest/example_notebooks/image_examples/image_classification/Explain%20an%20Intermediate%20Layer%20of%20VGG16%20on%20ImageNet.html (tikrinta 2024-06-06).

[MAP21] KT Yasas Mahima, Mohamed Ayoob ir Guhanathan Poravi. An Assessment of Robustness for Adversarial Attacks and Physical Distortions on Image Classification using Explainable AI. *AI-Cybersec@ SGAI*, p.p. 14–28, 2021.

[MMS+19] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras ir Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks, 2019. arXiv: 1706.06083 [stat.ML]. URL: https://arxiv.org/abs/1706.06083.

[MW+14] Sonali B Maind, Priyanka Wankar ir k.t. Research paper on basic of artificial neural network. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2(1):96–100, 2014.

[NAA+24] Md Nahiduzzaman, Lway Faisal Abdulrazak, Mohamed Arselene Ayari, Amith Khandakar ir SM Riazul Islam. A novel framework for lung cancer classification using lightweight convolutional neural networks and ridge extreme learning machine model with SHapley Additive exPlanations (SHAP). *Expert Systems with Applications*, 248:123392, 2024.

[ON15] Keiron O'shea ir Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[Par06] Irene Parrachino. Cooperative game theory and its application to natural, environmental and water resource issues, 2006.

[PKP18] Aakash Parmar, Rakesh Katariya ir Vatsal Patel. A review on random forest: An ensemble classifier. *International conference on intelligent data communication technologies and internet of things*, p.p. 758–763. Springer, 2018.

[PL16] Y.-S. Park ir S. Lek. Chapter 7 - Artificial Neural Networks: Multilayer Perceptron for Ecological Modeling. Sven Erik Jørgensen, redaktorius, *Ecological Model Types*. Tom. 28, Developments in Environmental Modelling, p.p. 123–140. Elsevier, 2016. DOI: https://doi.org/10.1016/B978-0-444-63623-2.00007-4. URL: https://www.sciencedirect.com/science/article/pii/B9780444636232000074.

[PMW+16] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha ir Ananthram Swami. Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks, 2016. arXiv: 1511.04508 [cs.CR].

[RDS+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause ir k.t. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. DOI: 10.1007/s11263-015-0816-y.

[Roj09] Raul Rojas. Neural Networks: a systematic introduction, 2009.

[RSG16] Marco Tulio Ribeiro, Sameer Singh ir Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier, 2016. arXiv: 1602.04938 [cs.LG].

[Sam23] Wojciech Samek. Chapter 2 - Explainable deep learning: concepts, methods, and new developments. Jenny Benois-Pineau, Romain Bourqui, Dragutin Petkovic ir Georges Quénot, redaktoriai, *Explainable Deep Learning AI*, p.p. 7–33. Academic Press, 2023. ISBN: 978-0-323-96098-4. DOI: https://doi.org/10.1016/B978-0-32-396098-4.00008-9. URL: https://www.sciencedirect.com/science/article/pii/B9780323960984000089.

[SCD+19] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh ir Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision*, 128(2):336–359, 2019-10. ISSN: 1573-1405. DOI: 10.1007/s11263-019-01228-7. URL: http://dx.doi.org/10.1007/s11263-019-01228-7.

[SD23] Jaydip Sen ir Subhasis Dasgupta. Adversarial Attacks on Image Classification Models: FGSM and Patch Attacks and their Impact, 2023. arXiv: 2307.02055 [cs.CV].

[SLS+22] Jiamei Sun, Sebastian Lapuschkin, Wojciech Samek ir Alexander Binder. Explain and improve: LRP-inference fine-tuning for image captioning models. *Information Fusion*, 77:233–246, 2022.

[SSA17] Sagar Sharma, Simone Sharma ir Anidhya Athaiya. Activation functions in neural networks. *Towards Data Sci*, 6(12):310–316, 2017.

[SWM17] Wojciech Samek, Thomas Wiegand ir Klaus-Robert Müller. Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models, 2017. arXiv: 1708.08296 [cs.AI].

[SZ15] Karen Simonyan ir Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, 2015. arXiv: 1409.1556 [cs.CV].

[TKL+21] Erzhena Tcydenova, Tae Woo Kim, Changhoon Lee ir Jong Hyuk Park. Detection of adversarial attacks in AI-based intrusion detection systems using explainable AI. *Human-Centric Comput Inform Sci*, 11, 2021.

[VFM04] Michael Van Lent, William Fisher ir Michael Mancuso. An explainable artificial intelligence system for small-unit tactical behavior. P.p. 900–907, 2004. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-9444223800%5C&partnerID=40%5C&md5=676e8c1489a82024551b5805d93d3947.

[ZBL+18] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen ir Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018.

[ZCZ+22] Hanxiao Zhang, Liang Chen, Minghui Zhang, Xiao Gu ir k.t. Interpretable Lung Cancer Diagnosis with Nodule Attribute Guidance and Online Model Debugging. *International Workshop on Interpretability of Machine Intelligence in Medical Image Computing*, p.p. 1–11. Springer, 2022.

[ZKL+16] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva ir Antonio Torralba. Learning deep features for discriminative localization. *Proceedings of the IEEE conference on computer vision and pattern recognition*, p.p. 2921–2929, 2016.

[ZQD+20] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong ir Qing He. A Comprehensive Survey on Transfer Learning, 2020. arXiv: 1911.02685 [cs.LG].

SANTRUMPOS

BIM: pagrindinis iteracinis metodas (angl. *Basic Iterative Method*),

DNT: dirbtinis neuroninis tinklas (angl. *artificial neural network*)

FGSM: greito gradiento ženklo metodas (angl. *fast gradient sign method*)

FPR: klaidingų teigiamų spėjimų dažnis (angl. *False Positive Rate*)

Grad-CAM: gradientais svertinis klasės aktyvacijos žemėlapis (angl. *Gradient-weighted Class Activation Mapping*)

IDS: įsibrovimo aptikimo sistema (angl. *Intrusion Detection System*)

ILSVRC: *ImageNet* didelio masto vaizdinių atpažinimo konkursas (angl. *ImageNet Large Scale Visual Recognition Challenge*)

KNT: konvoliucinis neuroninis tinklas (angl. *convolutional neural network*)

LIME: vietiniai interpretuojami modelių agnostiniai paaiškinimai (angl. *Local Interpretable Model-Agnostic Explanations*)

LRP: reikšmingumo sklaida sluoksniais (angl. *Layer-wise Relevance Propagation*)

PGD: projekcinio gradientinio nusileidimo metodas (angl. *Projected Gradient Descent*)

SHAP: SHapley priedų paaiškinimai (angl. *SHapley Additive exPlanations*)

TPR: teisingų teigiamų spėjimų dažnis (angl. *True Positive Rate*)

UMAP: (angl. *Uniform Manifold Approximation and Projection*)

XAI: paaiškinamasis dirbtinis intelektas (angl. *explainable AI*)