**Kauno technologijos universitetas**

Informatikos fakultetas

# Automatinio pažeistų miško medžių lajų klasifikavimo iš UAV vaizdų sistema ir tyrimas

Baigiamasis magistro studijų projektas

---

Projektą parengė

**Tomas Poška**

Projektui vadovavo

**Prof. dr. Rytis Maskeliūnas**

---

**Kaunas, 2026**

**Kauno technologijos universitetas**

Informatikos fakultetas

# Automatinio pažeistų miško medžių lajų klasifikavimo iš UAV vaizdų sistema ir tyrimas

Baigiamasis magistro studijų projektas

Programų sistemų inžinerija (6211BX011)

Projektą parengė

**Tomas Poška**

Projektui vadovavo

**Prof. dr. Rytis Maskeliūnas**

Projektą recenzavo

**Prof. dr. R. Damaševičius**

**Kaunas, 2026**

**Kauno technologijos universitetas**

Informatikos fakultetas

Tomas Poška

# Automatinio pažeistų miško medžių lajų klasifikavimo iš UAV vaizdų sistema ir tyrimas

## Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdamas (-a) kitų asmenų autoriaus ar kitų teisių, laikydamasis (-i) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;

2. visi baigiamajame projekte pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena projekto dalis nėra plagijuota nuo spausdintinių ar elektroninių šaltinių, o visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąraše;

3. baigiamajame projekte tinkamai laikiausi asmens duomenų apsaugos reikalavimų, nenaudojau neskelbtinų ar konfidencialių duomenų be teisėto pagrindo, o jei juos naudojau, jie yra tinkamai nuasmeninti;

4. jei rengiant baigiamąjį projektą naudojausi dirbtinio intelekto (toliau – DI) ar kitais automatizuotais įrankiais, juos taikiau pagal Universitete nustatytą tvarką, nepažeisdamas (-a) akademinio sąžiningumo principų;

5. nesumokėjau ir nesu įsipareigojęs (-usi) mokėti jokių įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis jokiam fiziniam ar juridiniam asmeniui;

6. suprantu, kad išaiškėjus akademinio nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikoma atsakomybė pagal Universitete nustatytą tvarką ir galiu būti pašalintas (-a) iš Universiteto; akademinio nesąžiningumo atvejis gali būti nagrinėjamas ir po studijų baigimo, inicijuojant kvalifikacinio laipsnio atšaukimo procedūrą.

**Kauno technologijos universitetas**

Informatikos fakultetas

**Baigiamojo magistro projekto užduotis**

| | |
|---|---|
| Projekto tema | Automatinio pažeistų miško medžių lajų klasifikavimo iš UAV vaizdų sistema ir tyrimas |
| Reikalavimai ir sąlygos (tikslinti pavadinimą pagal poreikį) | |
| Užduotį pateikęs asmuo | (pareigos, vardas, pavardė, parašas) (data) |

Tomas Poška. Automatinio pažeistų miško medžių lajų klasifikavimo iš UAV vaizdų sistema ir tyrimas. Magistro studijų baigiamasis projektas / projektui vadovavo Prof. dr. Rytis Maskeliūnas; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir studijų krypčių grupė: Programų sistemų inžinerija.

Reikšminiai žodžiai: Medžių ligų nustatymas, miško ekosistemos, kompiuterinė rega, dirbtinis intelektas, konvoliuciniai neuroniniai tinklai, klasifikavimas.

Kaunas, 2026. 72 p.

## Santrauka

Miško medžių ligos gali sparčiai plisti ir sukelti miško ekosistemų sutrikimus, todėl labai svarbu kuo anksčiau aptikti ir nustatyti medžių pažeidimus. Vienas dažniausiai naudojamų medžių sveikatos vertinimo rodiklių yra lajos defoliacijos lygis. Tradicinis defoliacijos vertinimas yra atliekamas lauko ekspedicijų metu, kai specialistai vizualiai apžiūri kiekvieną medį ir priskiria defoliacijos klasę. Toks procesas yra brangus, lėtas ir subjektyvus, todėl neįmanoma padengti didelių miško plotų. Automatizuota sistema gebanti patikimai nustatyti medžių lajų pažeidimų lygį naudojant bepiločių orlaivių vaizdus, leistų padidinti stebimų plotų aprėptį ir pagreitinti stebėsenos procesą.

Šiame darbe nagrinėjamas kompiuterinės regos metodų taikymas automatizuotam miško medžių lajų pažeidimų nustatymui ir klasifikavimui. Tyrime analizuojami iš bepiločių orlaivių surinkti miško vaizdų duomenų rinkiniai, apimantys skirtingų defoliacijos lygių medžių lajas. Tyrimo metu atliekama susijusių tyrimų ir kitose srityse taikomų metodų analizė, vertinant jų pritaikomumą medžių lajų pažeidimų klasifikavimo uždaviniui spręsti. Problemai spręsti taikomi pažangūs giluminio mokymosi modeliai, įskaitant konvoliucinius neuroninius tinklus, vienos stadijos objektų detektorius ir transformerių pagrindu veikiančias architektūras. Eksperimentinėje tyrimo metu siekiama nustatyti, kurios kompiuterinės regos architektūros yra tinkamiausios problemai spręsti, lyginant jų tikslumą ir apdorojimo laiką. Taip pat siekiama nustatyti, ar dviejų stadijų sprendimas, kai atskirai atliekamas objektų aptikimas ir klasifikavimas, gali tiksliau nustatyti miško medžių pažeidimo lygį nei vienos stadijos modelis kartu aptinkantis ir klasifikuojantis. Gauti rezultatai prisideda prie automatizuotos miškų sveikatos stebėsenos plėtros, ankstyvesnio ligų nustatymo bei tvaresnės miškotvarkos sprendimų kūrimo.

Tomas Poška. System and Study for Automated Classification of Damaged Forest Tree Crowns from UAV Imagery. Master‘s Final Degree Project supervisor Prof. dr. Rytis Maskeliūnas; Faculty of Informatics, Kaunas University of Technology.

Study field and study field group: Software Engineering.

Keywords: Tree disease detection, forest ecosystems, computer vision, artificial intelligence, convolutional neural networks, CNN, classification.

Kaunas, 2026. 72 p.

**Summary**

Forest tree diseases can spread quickly and disturb forest ecosystems, which is why detecting and identifying tree damage at an early stage matters a great deal. Crown defoliation is among the indicators most often used to assess tree health. The conventional way of measuring it relies on field expeditions, during which specialists inspect each tree by eye and assign it a defoliation class. Such work is expensive, slow and subjective, which makes it impractical for covering large forest areas. If an automated system could reliably estimate the level of crown damage from images taken by unmanned aerial vehicles, the monitored area could be expanded and the whole process sped up considerably.

This work studies how computer vision methods can be used to detect and classify forest tree crown damage automatically. The datasets analyzed in the study consist of forest images captured by UAVs, with tree crowns showing different degrees of defoliation. Related research, along with methods used in other fields, is reviewed to evaluate how well these approaches might transfer to the task of tree crown damage classification. The problem is approached using modern deep learning models, including convolutional neural networks, one-stage object detectors, and architectures built on transformers. The experimental part of the study aims to determine which computer vision architectures are best suited for the task by comparing their accuracy and processing time. A second question addressed here is whether splitting the task into two stages, where the objects are first detected and then classified, yields a more accurate estimate of tree damage than a single-stage model that does both at once. The findings contribute to the wider effort to automate forest health monitoring, improve early disease detection, and support more sustainable forestry management decisions.

## Turinys


Lentelių sąrašas ..... 9
Paveikslų sąrašas ..... 11
Santrumpų ir terminų sąrašas ..... 12
Įvadas ..... 14
**1.** Literatūros apžvalga ..... 15
1.1. Probleminė sritis ..... 15
1.1.1. Rinkos tyrimas ..... 16
1.2. Egzistuojantys sprendimai ..... 16
1.2.1. Tyrimai ir taikyti sprendimai probleminėje srityje ..... 16
1.2.2. Tyrimai ir taikyti sprendimai panašios problematikos srityse ..... 18
1.2.3. Sprendimų apibendrinimas ..... 21
1.3. Kompiuterinės regos modelių architektūros ..... 22
1.3.1. Klasikiniai konvoliuciniai neuroniniai tinklai ..... 22
1.3.2. Vienos stadijos detektoriai ..... 24
1.3.3. Transformerių architektūros ..... 25
**2.** Projektinė dalis ..... 27
2.1. Reikalavimų specifikacija ..... 27
2.1.1. Projekto kūrimo pagrindas ..... 27
2.1.2. Sistemos tikslai ..... 27
2.1.3. Apribojimai sprendimui ..... 27
2.1.4. Esama padėtis ..... 27
2.1.5. Veiklos kontekstas ..... 28
2.1.6. Sistemos panaudojimo atvejai ..... 29
2.1.7. Funkciniai reikalavimai ir reikalavimai duomenims ..... 37
2.2. Nefunkciniai reikalavimai ..... 38
2.2.1. Reikalavimai sistemos išvaizdai ..... 38
2.2.2. Reikalavimai panaudojamumui ..... 40
2.2.3. Reikalavimai vykdymo charakteristikoms ..... 40
2.2.4. Reikalavimai veikimo sąlygoms ..... 41
2.2.5. Reikalavimai sistemos priežiūrai ..... 41
2.2.6. Reikalavimai saugumui ..... 41
2.2.7. Kultūriniai-politiniai reikalavimai ..... 42
2.2.8. Sistemos statinis vaizdas ..... 42
2.3. Paketų detalizavimas ..... 43
2.3.1. Paketas „Naudotojo sąsaja“ ..... 43
2.3.2. Paketas „Logikos valdymas“ ..... 43
2.3.3. Paketas „Klasifikatorius“ ..... 44
2.4. Sistemos dinaminis vaizdas ..... 45
2.5. Išdėstymo vaizdas ..... 52
2.6. Duomenų vaizdas ..... 53
2.7. Kokybė ..... 53
2.8. Sistemos realizacija ..... 53
**3.** Tyrimas ..... 56

3.1. Naudojamų modelių aprašymas ... 56
3.1.1. Objektų aptikimo modeliai ... 56
3.1.2. Dviejų stadijų architektūra ... 57
3.1.3. Objektų klasifikavimo modeliai ... 58
3.2. Duomenų paruošimas ir analizė ... 59
3.2.1. Duomenų rinkinio apžvalga ... 59
3.2.2. Duomenų augmentacija ir pirminis apdorojimas ... 60
3.3. Eksperimento sąlygos ir vertinimo metrikos ... 62
3.3.1. Hiperparametrai ir mokymo aplinka ... 62
3.3.2. Vertinimo metrikos ... 62
3.4. Rezultatai ir analizė ... 63
3.4.1. Vienos stadijos modelių rezultatai ... 63
3.4.2. Dviejų stadijų modelių rezultatai ... 64
3.4.3. Vienos stadijos ir dviejų stadijų architektūrų palyginimas ... 65
3.5. Rezultatų aptarimas ir išvados ... 67
Išvados ... 69
Literatūros sąrašas ... 70

## Lentelių sąrašas

**1 lentelė** PA „Prisijungti“ specifikacija ..... 29
**2 lentelė** PA „Registruotis“ specifikacija ..... 30
**3 lentelė** PA „Peržiūrėti sistemos aprašymą“ specifikacija ..... 30
**4 lentelė** PA „Peržiūrėti modelių aprašymus“ specifikacija ..... 30
**5 lentelė** PA „Peržiūrėti duomenų statistiką“ specifikacija ..... 31
**6 lentelė** PA „Pateikti duomenis klasifikavimui“ specifikacija ..... 31
**7 lentelė** PA „Valdyti asmeninius rezultatus“ specifikacija ..... 31
**8 lentelė** PA „Atšaukti procesą“ specifikacija ..... 32
**9 lentelė** PA „Valdyti visus rezultatus“ specifikacija ..... 32
**10 lentelė** PA „Peržiūrėti rezultatą“ specifikacija ..... 32
**11 lentelė** PA „Parsisiųsti rezultatą“ specifikacija ..... 33
**12 lentelė** PA „Ištrinti įrašą“ specifikacija ..... 33
**13 lentelė** PA „Įkelti naujus duomenis“ specifikacija ..... 33
**14 lentelė** PA „Valdyti asmeninius duomenis“ specifikacija ..... 34
**15 lentelė** PA „Valdyti visus duomenis“ specifikacija ..... 34
**16 lentelė** PA „Patvirtinti duomenis“ specifikacija ..... 34
**17 lentelė** PA „Redaguoti anotacijas“ specifikacija ..... 35
**18 lentelė** PA „Ištrinti duomenis“ specifikacija ..... 35
**19 lentelė** PA „Peržiūrėti naudotojus“ specifikacija ..... 35
**20 lentelė** PA „Pakviesti naudotoją“ specifikacija ..... 35
**21 lentelė** PA „Pašalinti naudotoją“ specifikacija ..... 36
**22 lentelė** PA „Atkurti naudotoją“ specifikacija ..... 36
**23 lentelė** Klasifikavimo algoritmo reikalavimas ..... 37
**24 lentelė** Duomenų apdorojimo reikalavimas ..... 37
**25 lentelė** Duomenų importavimo reikalavimas ..... 37
**26 lentelė** Modelio informacijos pateikimo reikalavimas ..... 37
**27 lentelė** Naudotojo sąsajos intuityvumo reikalavimas ..... 38
**28 lentelė** Sąsajos vientisumo reikalavimas ..... 39
**29 lentelė** Klaidų valdymo reikalavimas ..... 39
**30 lentelė** Laukimo pranešimų reikalavimas ..... 39
**31 lentelė** Naudojimosi paprastumo reikalavimas ..... 40
**32 lentelė** Kalbos reikalavimas ..... 40
**33 lentelė** Atsako greičio reikalavimas ..... 40
**34 lentelė** Naršyklių suderinamumo reikalavimas ..... 41
**35 lentelė** Modulinės architektūros reikalavimas ..... 41
**36 lentelė** Duomenų validavimo reikalavimas ..... 41
**37 lentelė** Apsaugos nuo kenkėjiško kodo reikalavimas ..... 41
**38 lentelė** Slapukų naudojimo reikalavimas ..... 42
**39 lentelė** Dalykinės kalbos reikalavimas ..... 42
**40 lentelė** Pradinis duomenų rinkinio klasių pasiskirstymas ..... 59
**41 lentelė** Duomenų rinkinio struktūra ..... 60
**42 lentelė** Duomenų rinkinio objektų pasiskirstymas atlikus duomenų apdorojimą ..... 60
**43 lentelė** Vienos stadijos modelių geriausi rezultatai ..... 63
**44 lentelė** Pirmosios stadijos aptikimo rezultatai naudojant YOLOv11m ..... 64

**45 lentelė** Klasifikavimo modelių rezultatai ..... 65
**46 lentelė** Geriausių vienos ir dviejų stadijų konfigūracijų tikslumo metrikų palyginimas ..... 66
**47 lentelė** Vienos ir dviejų stadijų architektūrų skaičiavimo resursų palyginimas ..... 67

## Paveikslų sąrašas

**1 pav.** Segmentavimo ir defoliacijos įvertinimų palyginimai pritaikius *RTCLE* ir *TSCLR* apmokymo modelius [12] ... 18
**2 pav.** Daugiastulpelinio konvoliucinio neuroninio tinklo struktūra [13] ... 19
**3 pav.** Žmonių minios tankio nustatymo sistemos struktūrinė schema [16] ... 20
**4 pav.** Medžių klasifikavimo sistemos struktūrinė schema [17] ... 21
**5 pav.** Bendroji konvoliucinio neuroninio tinklo architektūros schema [21] ... 23
**6 pav.** YOLO šeimos detektoriaus architektūros schema [28] ... 24
**7 pav.** Vaizdo transformerio architektūros schema [26] ... 25
**8 pav.** Veiklos kontekstas ... 28
**9 pav.** Panaudojimo atvejų diagrama ... 29
**10 pav.** Duomenų modelis ... 38
**11 pav.** Sistemos išskaidymas į paketus ... 43
**12 pav.** Paketo „Logikos valdymas“ klasių diagrama ... 44
**13 pav.** PA „Peržiūrėti duomenų statistiką“ sekų diagrama ... 45
**14 pav.** PA „Prisijungti“ sekų diagrama ... 45
**15 pav.** PA „Registruotis“ sekų diagrama ... 46
**16 pav.** PA „Pateikti failą klasifikavimui“ sekų diagrama ... 47
**17 pav.** PA „Valdyti rezultatus“ sekų diagrama ... 48
**18 pav.** PA „Įkelti duomenis“ sekų diagrama ... 49
**19 pav.** PA „Valdyti duomenis“ sekų diagrama ... 50
**20 pav.** PA „Valdyti naudotojus“ sekų diagrama ... 51
**21 pav.** PA „Pakviesti naudotoją“ sekų diagrama ... 52
**22 pav.** Sistemos išdėstymo vaizdas ... 52
**23 pav.** Duomenų modelio diagrama ... 53
**24 pav.** Klasifikavimo puslapis – po failo ir klasifikavimo modelio pasirinkimo ... 54
**25 pav.** Klasifikavimo rezultatų peržiūros puslapis – rezultatų suvestinė ... 55
**26 pav.** Klasifikavimo rezultatų peržiūros puslapis – vizualiai sužymėtų objektų peržiūra ... 55
**27 pav.** Dviejų stadijų klasifikavimo strategija ... 57
**28 pav.** Duomenų rinkinio klasių pasiskirstymas ... 59
**29 pav.** Duomenų rinkinio klasių pasiskirstymas po pirminio apdorojimo ... 61
**30 pav.** Duomenų rinkinio klasių pasiskirstymas tarp dalių po pirminio apdorojimo ... 61

## Santrumpų ir terminų sąrašas

**Santrumpos:**

**UAV** (angl. *unmanned aerial vehicle*) – bepiločiai orlaiviai, naudojami vaizdų rinkimui ir stebėsenai iš oro;

**CNN** (angl. *convolutional neural network*) – konvoliucinis neuroninis tinklas;

**ReLU** (angl. *rectified linear unit*) – aktyvacijos funkcija, įvedanti netiesiškumą į neuroninius tinklus;

**YOLO** (angl. *you only look once*) – vienos stadijos objektų aptikimo modelių architektūra;

**RT-DETR** (angl. *real-time detection transformer*) – realaus laiko objektų aptikimo transformerių architektūra;

**RF-DETR** (angl. *region-free detection transformer*) – objektų aptikimo transformerių architektūra be inkarinių dėžių;

**ViT** (angl. *vision transformer*) – transformerių architektūra vaizdų analizei.

**FPN** (angl. *feature pyramid network*) – požymių piramidės tinklas, naudojamas įvairaus mastelio požymiams sujungti;

**PAN** (angl. *path aggregation network*) – požymių agregavimo tinklas, stiprinantis informacijos perdavimą tarp sluoksnių;

**CSPDarknet** (angl. *cross stage partial darknet*) – YOLO modeliuose naudojama požymių išgavimo architektūra;

**mAP** (angl. *mean average precision*) – vidutinio tikslumo metrika objektų aptikimo modeliams vertinti;

**MCNN** (angl. *multi-column convolutional neural network*) – daugiastulpelinis konvoliucinis neuroninis tinklas;

**RTCLE** (angl. *real-time crown loss estimation*) – realaus laiko medžių lajų praradimo vertinimo modelis;

**TSCLR** (angl. *two-step crown loss ranking*) – dviejų etapų medžių lajų praradimo vertinimo metodas.

**Terminai:**

**Defoliacija** – priešlaikinis medžių lapų ar spyglių kritimas, naudojamas medžių sveikatos vertinimui;

**Laja** – viršutinė medžio dalis, sudaryta iš šakų ir lapijos;

**Defoliacijos klasė** – kategorija, apibrėžianti medžio lapijos praradimo lygį;

**Fotogrametrija** – objektų formos, matmenų ir padėties nustatymas iš fotografijų;

**Kompiuterinė rega** – dirbtinio intelekto sritis, analizuojanti ir interpretuojanti vaizdus bei vaizdo įrašus;

**Gilusis mokymasis** – mašininio mokymosi metodų grupė, paremta daugiasluoksniais neuroniniais tinklais;

**Objektų aptikimas** – procesas, kurio metu vaizde nustatomos objektų vietos ir jų klasės;

**Objektų klasifikavimas** – procesas, kurio metu objektas priskiriamas konkrečiai klasei;

**Segmentavimas** – vaizdo skaidymas į prasmingas sritis arba objektus;

**Anotacija** – objektų žymėjimas vaizduose, naudojamas modelių apmokymui;

**Duomenų augmentacija** – dirbtinis mokymo duomenų išplėtimas keičiant esamus vaizdus;

**Hiperparametrai** – prieš modelio mokymą nustatomi parametrai, valdantys mokymo procesą;

**Mokymo duomenų rinkinys** – duomenys, naudojami modelio apmokymui;

**Testavimo duomenų rinkinys** – duomenys, naudojami modelio veikimui įvertinti;

**Vienos stadijos detektorius** – objektų aptikimo modelis, vienu metu atliekantis aptikimą ir klasifikavimą;

**Dviejų stadijų architektūra** – objektų aptikimo metodas, kuriame pirmiausia aptinkami objektų regionai, o vėliau jie klasifikuojami;

**Dėmesio mechanizmas** (angl. *attention mechanism*) – metodas, leidžiantis modeliui nustatyti svarbiausias įvesties dalis;

**Savidėmesys** (angl. *self-attention*) – dėmesio mechanizmo tipas, modeliuojantis ryšius tarp visų įvesties elementų;

**Požymių žemėlapis** (angl. *feature map*) – konvoliucinio sluoksnio išvestis, aprašanti aptiktus požymius;

**Ribojantis stačiakampis** (angl. *bounding box*) – stačiakampė sritis, apibrėžianti objekto vietą vaizde;

**Perteklinis pritaikymas** (angl. *overfitting*) – reiškinys, kai modelis per daug prisitaiko prie mokymo duomenų ir blogiau apibendrina naujus duomenis.

# Įvadas

Įvairios medžių ligos, gali atsirasti dėl įvairių priežasčių ir plisti labai sparčiai, ypač jei tam tikros medžių rūšys yra dominuojančios miško plote. Sistema, kuri leistų greitai ir efektyviai aptikti medžių ligas ankstyvose jų stadijose būtų labai naudinga miškotvarkoje – padėtų išsaugoti miškų sveikatą ir išvengti galimų ekosistemų sutrikimų. Medžio laja yra pagrindinis miško medžių komponentas, kuriuo galima nustatyti medžio sveikatos būklę [1]. Medžio lajos defoliacija yra dažniausiai naudojamas matas nustatyti medžio sveikatą [2]. Sistema, gebanti automatiškai klasifikuoti medžių defoliacijos lygį, galėtų būti naudinga aptikti ankstyvuosius medžių ligų požymius, apie juos signalizuoti miškotvarkos specialistams.

Projekto metu bus kuriama sistema, kuri padės nustatyti miško medžių defoliacijos lygį. Sistemai įgyvendinti bus kuriama sistema, kurios pagalba naudotojai galės nufotografuoti miško medžius ir įvertinti jų defoliacijos lygį, gauti atsaką iš sistemos apie įvertinimo tikslumą. Tokia sistema padės surinkti didelį duomenų kiekį, kurių pagalba apmokant konvoliucinių neuroninių tinklų algoritmus bus siekiama efektyviau ir tiksliau nustatyti medžių lajų defoliacijos lygį.

Tikslas – sukurti automatinę sistemą, kuri iš bepiločių orlaivių nuotraukų patikimai klasifikuotų miško medžių lajų defoliciacijos lygį.

Uždaviniai:
1. išanalizuoti miško medžių lajų pažeidimų nustatymo problematiką ir egzistuojančius kompiuterinės regos sprendimus;
2. išanalizuoti objektų aptikimo ir klasifikavimo modelių architektūras;
3. parinkti ir apmokyti vienos stadijos objektų aptikimo modelius medžių pažeidimų klasifikavimo užduočiai;
4. įgyvendinti ir apmokyti dviejų stadijų klasifikavimo architektūrą, sujungiančią objektų aptikimo ir klasifikavimo modelius;
5. įvertinti ir palyginti vienos bei dviejų stadijų architektūrų klasifikavimo tikslumą.

# 1. Literatūros apžvalga

## 1.1. Probleminė sritis

Aptikti miško medžių ligas ankstyvojoje stadijoje yra svarbi, bet sudėtinga užduotis. Medžio laja yra pagrindinis miško medžių komponentas, kuriuo galima nustatyti medžio sveikatos būklę [1]. Medžio lajos defoliacija yra dažniausiai naudojamas matas nustatyti medžio sveikatą [2].

Lajų defoliacija vertinama 5 % gradacija vizualiai nustatant medžio lapijos nuostolius, lyginant juos su etaloniniu medžiu. Visiškai sveiko (turinčio visą lapiją) medžio defoliacija prilyginama 0 %, o nudžiūvusio – 100 %. Pagal defoliaciją medžiai skirstomi į 5 klases [3]:

1. santykinai sveiki medžiai (defoliacija 0–10 %);
2. menkos defoliacijos (15–25 %);
3. vidutinės defoliacijos (30–60 %);
4. didelės defoliacijos (65–95 %);
5. žuvę medžiai (100 %).

Medžio lapų defoliaciją gali lemti skirtingos priežastys, įskaitant ligas, kenkėjus, oro sąlygas, todėl yra svarbu identifikuoti defoliacijos stadijas ir jos progresavimą, siekiant veiksmingai įsikišti, nustatant konkrečias priežastis. Medžio defoliaciją gali lemti daug faktorių. Didelę įtaką medžių viršūnių būklei turi natūralūs veiksniai pvz. medžių senėjimas [3]. Todėl yra netikslinga mėginti paaiškinti defoliaciją remiantis tik vienu veiksniu, pavyzdžiui, oro tarša, be papildomų įrodymų. Aplinkos sąlygų įtakos ir toleruotino defoliacijos lygio negalima tiksliai įvertinti kiekybiškai, pažeistų medžių negalima atskirti nuo sveikų tik pagal tam tikrą defoliacijos lygio slenkstį, t.y. 25 % medžio defoliacijos riba nebūtinai nustato, kad medis yra pažeistas fiziologiškai [4]. Kita problema, nustatant medžių sveikatos būklę vadovaujantis defoliacija yra žmogiškasis faktorius. Medžio viršūnės tankumas yra nespecifiškas simptomas, kuris negali neginčytinai paaiškinti faktinių miško pažeidimų priežasčių. Kadangi, lajų sveikatos vertinimai yra subjektyvūs ir charakteristikos yra grubios, gali atsirasti rimtų klaidų ir klausimų dėl miško sveikatos tendencijų patikimumo [5]. Taigi, nustatyti medžių sveikatos būklę yra sudėtinga užduotis, reikalaujanti detalaus stebėjimo ir įvairiapusės analizės.

Vis daugiau tyrimų yra atliekama siekiant kompiuterizuoti miško medžių sveikatos įvertinimo problemą, siekiant pigiau, efektyviau ir tiksliau nustatyti medžių ligas. Fotogrametrija ir nuotoliniai jutikliai yra papildomos priemonės šiuolaikinėje miškininkystėje, kurios yra dažnai naudojamos miško vertinimuose, analizuojant medynų augaliją, stebint gamtos nelaimės atvejus, matuojant medžio viršūnių dangos plotą, biologinę įvairovę ir kituose panaudojimo atvejuose [6]. Fotogrametrinių ir nuotolinio stebėjimo duomenų analizę galima atlikti automatiškai, naudojant kompiuterinę regą ar gilųjį mokymą. Yra atliktų tyrimų ir sprendimų, kurie pasitelkia mašininį mokymą kaip klasifikavimo įrankį medžių sveikatos vertinime miškuose. Tačiau, kompiuterizuojant šią problemą, t.y. siekiant automatizuoti pažeistų medžių lajų klasifikavimą susiduriama su šiomis galimomis problemomis:

- sudėtingas medžio atskyrimas nuo aplinkos (kitų medžių);
- medžiais skiriasi rūšimis, aukščiu, forma, spalva;
- skalės nustatymas;

- medžio lapų tankumas nuotraukoje gali skirtis nuo kampo, kuriuo buvo fotografuojama [7];
- sezoniškumas – medžių išvaizda gali keistis skirtingais metų laikais.

Kitos problemos minimos anksčiau atliktuose sprendimuose yra duomenų trūkumas. Skirtinguose regionuose auga skirtingų rūšių medžiai, tai sunkina duomenų kaupimą kiekvienai skirtingai medžių rūšiai.

### 1.1.1. Rinkos tyrimas

Šiuo metu rinkoje nėra sistemų skirtų nustatyti medžių lajų defoliacijos lygį. Analizuojant mokslininkų atliktus defoliacijos nustatymo tyrimus [8] yra pateikiamos kelios mobilios sistemos kaip geri pavyzdžiai kuriamai sistemai – „Pl@ntNet“ [9], “LeafSnap“ [10] ir “iNaturalist“ [11]. Programėlės-sistemos yra skirtos identifikuoti, klasifikuoti įvairius augalus, taigi sprendžia panašia problemą. „Pl@ntNet“ programėlė gali klasifikuoti augalus, nustatyti medžio rūšį iš pateiktos nuotraukos. “iNaturalist“ programėlė leidžia nufotografuoti augalų ar gyvūnų rūšis, priskirti lokaciją kurioje užfiksuota nuotrauka ir pateikti nuotraukos aprašymą. “LeafSnap“ programėlė leidžia identifikuoti augalus iš lapų nuotraukų. Kuriama sistema, būtų labiau specializuota pateiktų pavyzdžių versija t.y. panašios paskirties kaip „Pl@ntNet“ programėlė, tačiau skirta lajų defoliacijai nustatyti.

„Pl@ntNet“ programėlė „Google Play“ platformoje turi daugiau nei 10 milijonų parsisiuntimų, “LeafSnap“ ir “iNaturalist“ po daugiau nei milijoną. Programėlės turi *iOS* versijas, taigi parsisiuntimų skaičius tikėtina yra dar didesnis. Tai rodo vartotojų, tyrėjų-mėgėjų susidomėjimą augalų identifikavimo programomis, galima teigti, kad rinkoje yra paklausa panašios paskirties sistemoms.

## 1.2. Egzistuojantys sprendimai

### 1.2.1. Tyrimai ir taikyti sprendimai probleminėje srityje

#### 1.2.1.1. Defoliacijos vertinimas iš antžeminių nuotraukų

„Miško medžių defoliacijos įvertinimas naudojant nuotraukas padarytas nuo žemės“. [8] Tyrimas buvo atliekamas siekiant efektyviau atlikti miško medžių defoliacijos įvertinimą, interpretuojant tai kaip vaizdo interpretavimo iššūkį. Klasifikavimo sprendime buvo naudojami konvoliuciniai neuroniniai tinklai (angl. *convolutional neural networks, CNN*). Tyrimo eiga [8]:

1. ekspertai aplanko atrinktus medžius skirtinguose miškuose;
2. padaro po vieną kiekvieno medžio spalvotą nuotrauką naudodami komercinę kamerą;
3. kiekvienam medžiui priskiria defoliacijos įvertinimą;
4. visos surinktos nuotraukos yra apkarpomos į kvadratą, paverčiant jas galimomis apmokyti CNN modelį;
5. atliekamas nuotraukų apdorojimas pvz. nuotraukų apsukimas.
6. CNN modelis yra apmokomas naudojant įvertintų ir apdorotų nuotraukų poaibį;
7. modelis ištestuojamas palyginant numatytus rezultatus su ekspertų įvertinimais.

Problemai spręsti buvo apmokomas ResNet konvoliucinių neuroninių tinklų modelis. Apmokant CNN modelį nebuvo grupuojamos nuotraukos pagal medžio rūšį, t.y. visos turimos nuotraukos buvo sprendimas įvertina vieną sceną-nuotrauką – iš jos nustato medžio defoliacijos lygį. Prieš apmokant CNN modelį nuotraukos nebuvo apdorotos. CNN modelis buvo apmokytas su 2108 nuotraukomis, autorių teigimu tai yra per mažas kiekis duomenų, kad būtų galima apmokyti algoritmą pagal atskiras medžių rūšis. Taip pat, visos nuotraukos buvo atliktos ir įvertintos nuo liepos pradžios iki rugpjūčio pabaigos, kad atsiriboti nuo sezoniškumo.

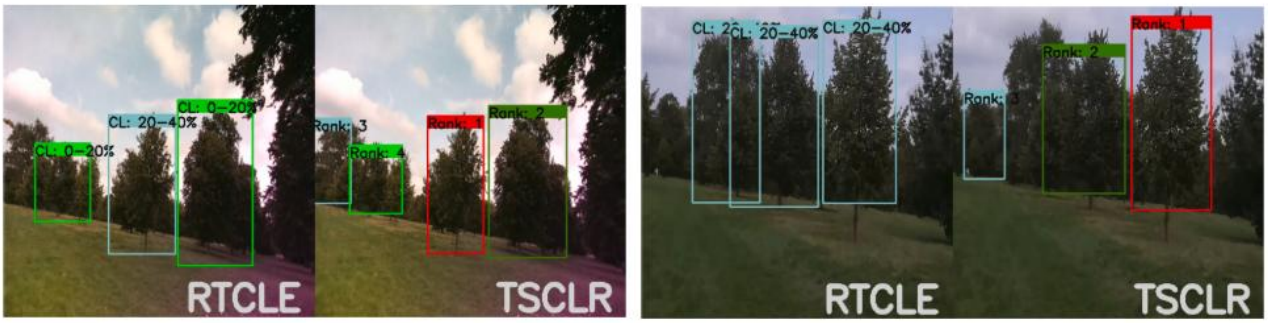
Tyrimo autorių teigimu taikytas sprendimas yra perspektyvus. „Kaip ir su dauguma (giliųjų) mašininio mokymosi metodų, modelis pasirodo gerai visose situacijose, turint pakankamai mokymo duomenų, tuo tarpu, našumas kenčia, jei mokymo pavyzdžių skaičius yra labai mažas.“ [8]

Tyrimo autoriai teigia, kad sukurtas modelis nustato medžių defoliacijos lygį tik su 8 % paklaida lyginant su žmonių ekspertų vertinimu. Nepavykę, prasčiausiai klasifikuoti atvejai buvo su aukštos defoliacijos arba mirusio medžio nuotraukomis, autorių pateikiami sudėtingų atvejų nuotraukose. Pateiktuose pavyzdžiuose modelis aptiko daug žemesnį defoliacijos lygį, nei buvo įvertinta ekspertų. Autorių manymu taip atsitiko, nes klasifikatorius fone esančių medžių lapus priskirdavo negyvam medžiui. Tyrimo metu nuotraukų su labai aukšta defoliacija buvo mažai, todėl, autorių teigimu, su daugiau ir tikslesniais duomenimis modelis galėtų duoti geresnius rezultatus. Vieną iš problemų įžvelgia, reiktų surinkti daugiau kokybiškų nuotraukų, duomenų, kuriose dominantis objektas, medis yra gerai atskirtas nuo aplinkos. Tyrėjai pateikia dvi gaires, kurios gali pagerinti problemos sprendimo rezultatus ateityje ir įgalintų sprendimą pritaikyti praktikoje. Pirma, siekiant pagerinti CNN apibendrinimo galimybes, reikėtų surinkti daugiau kokybiškų duomenų. Antra, naudojant mobiliasias programėles ir minios-piliečių mokymo projektą, būtų įmanoma surinkti didesnį duomenų kiekį iš įvairesnių ir platesnių teritorijų. [8]

#### 1.2.1.2. Lajų pažeidimų vertinimas iš UAV vaizdų taikant YOLO ir sintetinius duomenis

„Vaizdu grindžiamas medžių viršūnių praradimo įvertinimas atskiriems medžiams naudojant nuotolinio orlaivio robotus.“ Atlikto eksperimento tikslas ištirti, ar sintetiškai sukurtos medžių nuotraukos gali būti naudojamos mokyti *RTCLE* (angl. *akronimas real-time crown loss estimation*) modelį, kadangi sunku surinkti didelį nuotraukų-duomenų kiekį su ekspertų įvertinimais.[12] Sprendimas yra taikomas aptikti medžių defoliaciją gaunant duomenis realiu laiku iš drono skrydžio. Taikytas sprendimas naudojo *WSL* duomenų rinkinį (tie patys duomenys naudoti pirmame pavyzdyje) ir simuliacinius-sintetinius duomenis, 3D sugeneruotus vaizdus. 90 % duomenų buvo 3D sugeneruoti vaizdai ir 10 % duomenų tikros nuotraukos su ekspertų įvertintais defoliacijos lygiais. Naudojant sintetinius duomenis buvo renkamos nuotraukos iš skirtingų kampų, kameros aukščio, medis sukamas aplink savo ašį – taip vienas modelis sugeneruoja įvairius duomenų pavyzdžius. Tokio metodo pritaikymas įgalina sugeneruoti daug duomenų, nes vaizdai yra skaitmenizuoti – turint arba sukūrus skirtingus medžių modelius galima surinkti sudėtingų atvejų duomenų, gauti vaizdus iš tikslių kampų, atstumų.

Sprendimas paremtas segmentacija ir klasifikavimu naudojant *YOLO* (*you only look once*) kompiuterinės regos algoritmą. Tyrimo metu buvo sukurti du modeliai: Pirmame, medžiai klasifikuojami į standartines 5 kategorijas kas 20 % pagal defoliacijos lygį. Antrame (*TSCLR* (angl. akronimas *two-step crown loss ranking*)), medžiai klasifikuojami pagal santykinį lajų nuostolį lyginant vienus su kitais (žr. 1 pav.).




**1 pav.** Segmentavimo ir defoliacijos įvertinimų palyginimai pritaikius *RTCLE* ir *TSCLR* apmokymo modelius [12]

Tyrimo autoriai teigia, kad apmokant RTCLE modelį vien sintetiniais duomenimis galima pasiekti pakankamai gerą medžių defoliacijos vertinimo efektyvumą. Apmokius RTCLE modelį kartu su ekspertų įvertintomis nuotraukomis pasiekti rezultatai yra geresni nei apmokius tik ekspertų įvertintais duomenimis, vertinant nuotraukas neįtrauktas į mokymo duomenų rinkinį. Objektyvus palyginimas tarp sukurtų modelių medžių defoliacijos nustatymo tikslumo lyginant su WSL ekspertų įvertintais duomenimis nebuvo pateiktas, nes tyrimas nustato defoliacijos lygį kiekvienam medžiui atskirai, o ekspertų įvertintuose duomenyse yra pateikiamas scenos, bendras nuotraukos vaizdo defoliacijos lygis.

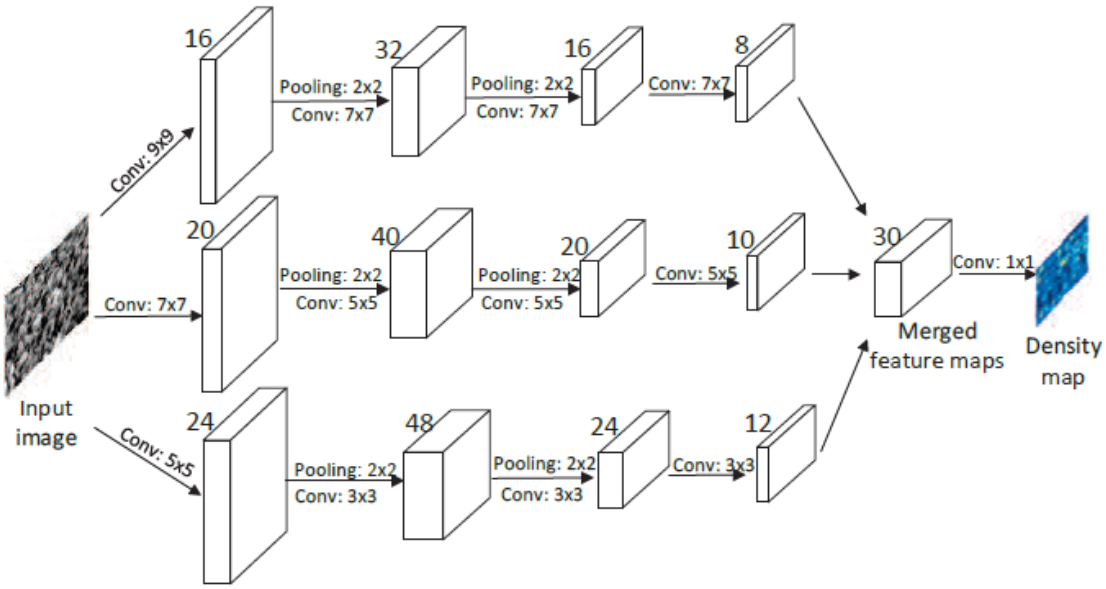
Tyrėjai apibendrina, kad kompiuterinės regos algoritmas gali būti efektyvus, patenkinamai klasifikuoti apmokytas vien tik sintetiniu duomenų rinkiniu. Tačiau, norint sukurti patikimą klasifikavimo modelį, neužtenka vien tik sintetinių ir/ar mažo kiekio realių duomenų. Todėl, teigia, kad norint sukurti patikima modelį reikalingas didelis kiekis kokybiškų ir įvairių duomenų, kartu sujungus sintetinius duomenis galima pasiekti aukštą našumą.[12] Sintetinių duomenų problema yra patikimam defoliacijos nustatymui reikalingas probleminės srities ekspertų įvertinimas. Siekiant tobulinti modelio efektyvumą, siūlo papildyti RTCLE modelio mokymo duomenų rinkinį įvairesniais vaizdais, naudojant tiek sintetinius, tiek realius duomenis, o taip pat išplėsti medžių rūšių bei geometrijų generavimą. Pateikia siūlymą, sintetiniais duomenimis sukurti sudėtingų atvejų vaizdų, kurių ypač trūksta nuotraukų rinkinyje.

### 1.2.2. Tyrimai ir taikyti sprendimai panašios problematikos srityse

#### 1.2.2.1. Minios tankio nustatymas naudojant daugiastulpelinį konvoliucinį neuroninį tinklą

Panašių sričių sprendimai, kurių taikytais metodais galima eksperimentuoti gali būti žmonių kiekio minioje nustatymas. Galima rasti daug įvairių sprendimų siekiančių nustatyti žmonių kiekį minioje [13][14][15], kurių taikomų metodų įgyvendinimas gali būti naudingas kuriant modelį skirtą medžių lajų tankio nustatymui. Pavyzdžiui, tyrimas žmonių kiekio minioje nustatymui iš vienos nuotraukos naudojant daugiastulpelinį konvoliucinį neuroninį tinklą.[13]

Tyrimas siekia sukurti metodą, kuris gali tiksliai įvertinti minios skaičių iš atskiros nuotraukos, turinčios įvairų minios tankį ir perspektyvą. Tyrimo autoriai sprendžia kelias problemas aprašytas probleminėje srityje (žr. 1.1 sk.), t.y. skalės įvertinimą, minios atskyrimą tolumoje nuo arti kameros žmonių esančių. „Žmonių skaičiaus minioje nustatymas yra sudėtinga užduotis dėl okliuzijos, žmonių galvų mastelio variacijų ir nelygių žmonių pasiskirstymų.“ [15] Problemai spręsti tyrimo autoriai sukūrė *MCNN* (angl. *Multi-column Convolutional Neural Network*) *CNN* modelį (žr. 2 pav.), gebantį nustatyti skirtingus minių tankius vienoje nuotraukoje. Sukurtas modelis nebando skaičiuoti kiekvieno žmogaus galvos nuotraukoje, o sukuria minios tankio žemėlapį (pavyzdžiui, kiek žmonių viename kvadratiniame metre).




**2 pav.** Daugiastulpelinio konvoliucinio neuroninio tinklo struktūra [13]

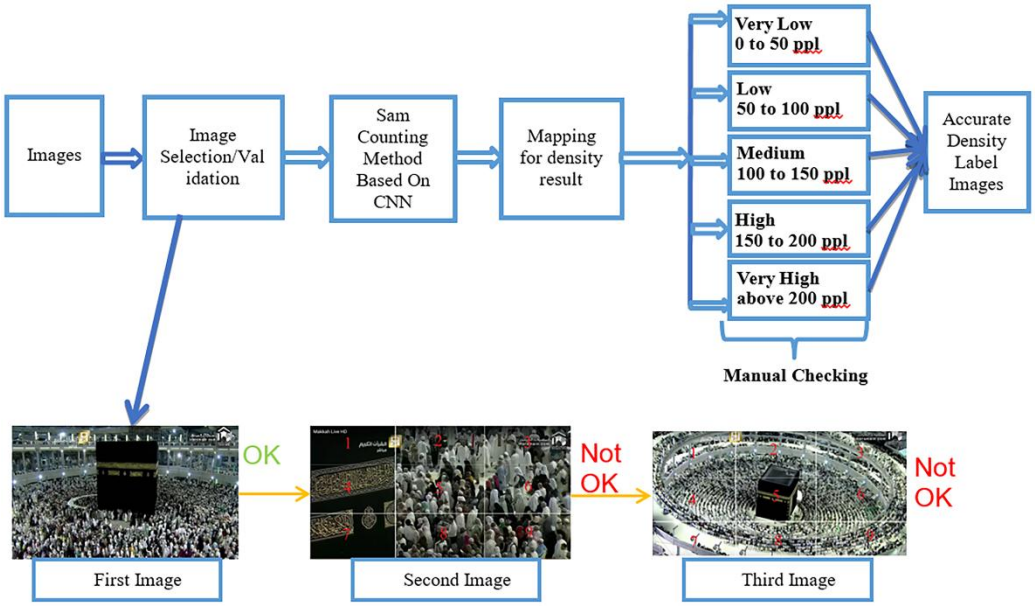
*MCNN* architektūra leidžia naudoti įvairaus dydžio ar rezoliucijos įvesties paveikslėlius. Naudojant filtrus su skirtingo dydžio priimančiomis sritimis (angl. *receptive fields*), kiekvieno stulpelinio CNN išmoksta savybes, prisitaikančias prie žmonių/galvų dydžio kaitos dėl perspektyvos poveikio arba paveikslėlio rezoliucijos.

Tyrimo autoriai teigia, kad daugiastulpelinio konvoliucinio neuroninio tinklo modelis geba nustatyti tikslų minios tankumą iš įvairių perspektyvų. Tyrėjų teigimu, MMNet modelis, pasižymi labai dideliu efektyvumu ir puikiai susitvarko su skirtingomis minios tankio variacijomis.

#### 1.2.2.2. Minios tankio klasifikavimas taikant duomenimis orientuotą metodą

Minios tankio klasifikavimo sprendimas [16]. Tyrimo autoriai aprašo jų sprendimo metodą pritaikius konvoliuciniu neuroninius tinklus sprendžiant minios tankio nustatymo, klasifikavimo problemą. Šis sprendimas yra aktualus, nes sprendžia panašią problemą susijusią su tankio nustatymu. Sprendžiant problemą duomenys yra klasifikuojami į keturias kategorijas, priklausomai nuo žmonių tankio aikštėje. Taikytas sprendimas yra orientuotas į duomenis (angl. *data-centric*). Prieš mokant nuotraukos yra atrenkamos, kad dominanti

erdvė būtų nuotraukos centre ir tinkamame nuotolyje t.y. modelis apmokomas tik su nuotraukomis atitinkančiomis fotografuotą kampą, priartinimą (žr. 3 pav.).



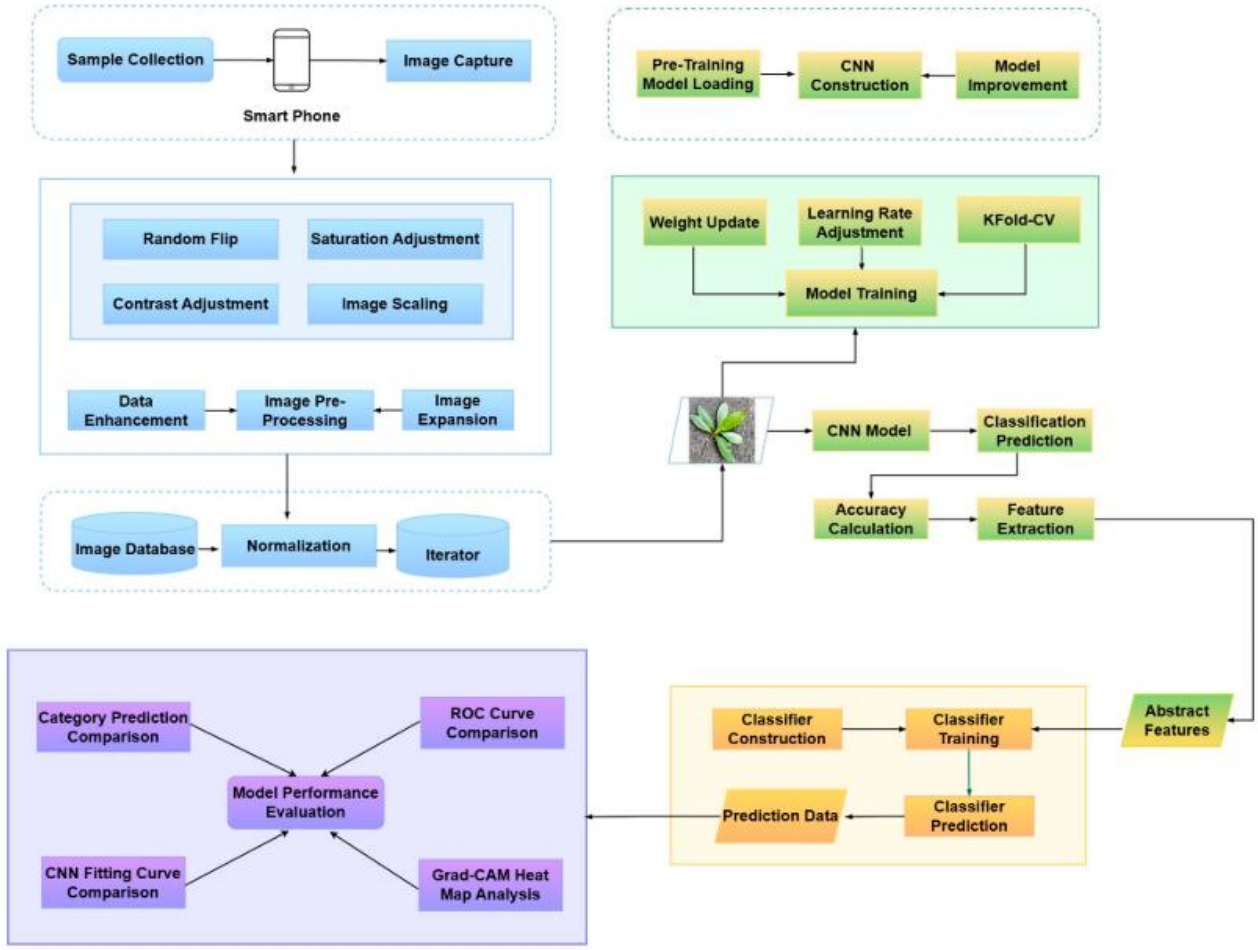

**3 pav.** Žmonių minios tankio nustatymo sistemos struktūrinė schema [16]

Autoriai teigia, kad jų sukurtas modelis geba nustatyti per daug tankiai užpildytą erdvę 100 % tikslumu naudojant jų duomenų rinkinį ir 97 % tikslumu naudojant ResNet neuroninių tinklų modelį. Tyrimas demonstruoja, kad duomenimis paremtas klasifikavimo sprendimas buvo efektyvus panašios problematikos užduotyje.

#### 1.2.2.3. Medžių rūšių atpažinimas iš lapų nuotraukų taikant gilųjį mokymąsi

"Lapų ir šakelių atpažinimas naudojant gilųjį mokymąsi ir kombinuotus klasifikavimo algoritmus." [17] Tyrimas sprendžia medžių rūšių klasifikavimo, automatinio atpažinimo problemą naudojant nuotraukas. Sprendimas nustato medžių rūšis naudojant atskiras lapų nuotraukas. Tyrimo eiga paprasta (žr. 4 pav.):

1. renkami duomenys-nuotraukos, jos apdorojamos geresniam pritaikymui – atliekama skalės, kontrasto standartizacija;
2. apmokomi konvoliuciniai neuroniniai tinklai;
3. klasifikavimo modeliai ištestuojami naudojant duomenis nenaudotus mokymo metu;
4. atliekama rezultatų analizė.




**4 pav.** Medžių klasifikavimo sistemos struktūrinė schema [17]

Surinkti ir apdoroti duomenys buvo apmokinti skirtingais neuroninių tinklų modeliais. Geriausias tirtas modelis pasiekė 99,86 % tikslumą atpažįstant 21 medžių rūšis. Autorių įgyvendinta sistema geba efektyviai klasifikuoti medžių rūšis naudojant lapų, šakų nuotraukas. Siekiant tobulinti modelį, tyrėjai įvardina panašias problematikas ir pasiūlymus kaip pirmo probleminės srities tyrimo autoriai (žr. 1.2.1.1 sk.). Tyrėjai iškelia poreikį plėsti medžių rūšių skaičių duomenų rinkinyje, įtraukti daugiau medžių savybių, siekiant geriau atpažinti ir įvertinti tų rūšių bruožus. Taip pat, siekiant gauti geresnius rezultatus ir efektyvesnį mokymąsi siūlo sukurti medžių rūšių identifikavimo mobilią programėlę.

### 1.2.3. Sprendimų apibendrinimas

Probleminės srities sprendimų autoriai įvardija duomenų trūkumą kaip pagrindinę problemą ir teigia, kad daugiau kokybiškai sužymėtų duomenų pagerintų neuroninių tinklų kokybę. Duomenys yra pagrindinis faktorius, lemiantis konvoliucinių neuroninių tinklų *CNN* veikimą. Efektyvus duomenų rinkimas ir tinkamas jų apdorojimas yra esminiai žingsniai siekiant sukurti patikimus ir tikslius modelius. Modelio veikimas galiausiai yra apribotas mokymo duomenų kokybe. Tačiau gauti aukštos kokybės mokymo duomenų yra ilgas ir brangus procesas. [18]. Pagrindinės įvardintos sritys, kurios gali teigiamai paveikti klasifikavimo modelį:

- surinkti daugiau ekspertų įvertintų nuotraukų;

- surinkti įvairesnių, sudėtingų atvejų duomenų;
- galimas sintetinių duomenų panaudojimas;
- modelio apmokymas atskiriant skirtingus medžius;
- mobilios programėlės skirtos duomenų rinkimui kūrimas.

Tyrimuose nebuvo nustatomas medžio tipas, tyrimų autoriai teigia, kad turint pakankamai duomenų, kad galima būtų apmokyti neuroninius tinklus atskiriant medžių rūšis, tai gali reikšmingai pagerinti klasifikavimo kokybę. Antrame panašių sprendimų pavyzdyje (žr. 1.2.2.2 sk.) autoriai teigia pasiekę labai aukštą klasifikavimo tikslumą taikydami į duomenis orientuotą metodologiją. Trečiame panašios problematikos pavyzdyje (žr. 1.2.2.3 sk.) pateikiamas pavyzdys gebantis efektyviai klasifikuoti medžių rūšis naudojant medžių lapų ir šakelių nuotraukas. Tyrimo autoriai įvardija tas pačias problemas siekiant tobulinti klasifikavimą. Siekiant surinkti našų ir įvairų duomenų rinkinį egzistuoja kitos, natūralių veiksnių problemos, kurios gali lemti modelio netikslumus:

- skirtingos medžių rūšys;
- medžiai keičia lapų spalvą dėl sezoniškumo;
- medžiai praranda lapus žiemos metu;
- apsnigtos medžių šakos;
- kt.

Šioms problemoms spresti analizuotų sprendimų autoriai naudojo, rinko duomenis pasirinktais mėnesiais – šiltuoju metų laiku. Tačiau, teigia, siekiant sukurti patikimą modelį, kurį galima būtų pritaikyti praktiškai, šių problemų sprendimas yra aktualus.

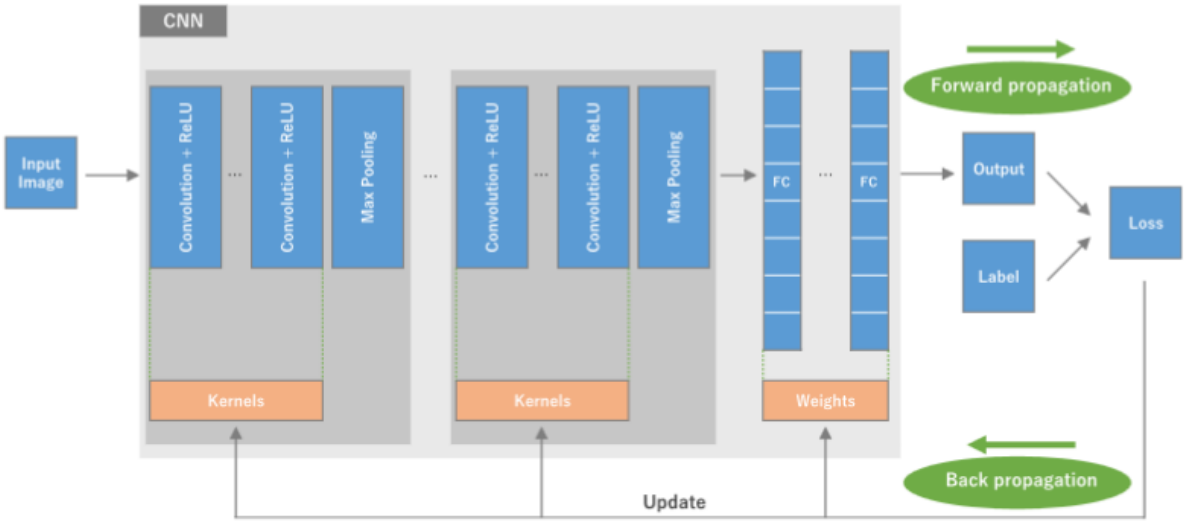
## 1.3. Kompiuterinės regos modelių architektūros

Šiame skyriuje apžvelgiamos trys pagrindinės kompiuterinės regos modelių kryptys: konvoliucinių neuroninių tinklų (CNN) architektūros, vienos stadijos objektų detektoriai ir transformerių architektūros.

### 1.3.1. Klasikiniai konvoliuciniai neuroniniai tinklai

Konvoliuciniai neuroniniai tinklai (CNN) – viena plačiausiai taikomų giluminio mokymosi architektūrų, ypač tinkama darbui su erdvinę struktūrą turinčiais duomenimis, pavyzdžiui, vaizdais ar vaizdo įrašais [19]. Pagrindinė CNN idėja – hierarchinis požymių išgavimas: ankstyvuosiuose sluoksniuose tinklas atpažįsta paprastus erdvinius dėsningumus: kraštines, tekstūras, spalvinius perėjimus, o gilesniuose sluoksniuose šie požymiai sujungiami į sudėtingesnes reprezentacijas, atitinkančias objektų dalis ar visus objektus [20]. Tipinę CNN architektūrą sudaro trys pagrindiniai sluoksniai (žr. 5 pav.). Pirmasis, konvoliucinis sluoksnis yra pagrindinis tinklo elementas, kuris per vaizdą „slenka" mažus filtrus (angl. *kernels*), kurie ieško tam tikrų vizualinių elementų, pavyzdžiui, kraštinių, kampų ar tekstūrų. Šių filtrų reikšmės nėra iš anksto nustatytos, tinklas jas išmoksta treniravimo metu, palaipsniui prisitaikydamas atpažinti vis tinkamesnius požymius. Po konvoliucinio sluoksnio dažniausiai taikoma aktyvacijos funkcija ReLU (angl. *Rectified Linear Unit*), kuri įveda netiesiškumą – be jos tinklas galėtų atpažinti tik paprastus, tiesinius ryšius ir nesugebėtų išmokti sudėtingų vaizdo struktūrų. Antrasis, telkimo sluoksnis (angl. *pooling layer*) sumažina vaizdo dydį, kiekvieną nedidelį pikselių regioną pakeisdamas viena reikšme, dažniausiai didžiausia (angl. *max pooling*) arba vidutine (angl. *average pooling*).

Toks suspaudimas turi kelias naudas – tinklas dirba greičiau, naudoja mažiau atminties, mažiau persimoko, taip pat tampa atsparesnis nedideliems objekto poslinkiams vaizde (angl. *translation invariance*), todėl objektas atpažįstamas net jei vaizde jis pasislinkęs keliais pikseliais. Trečiasis, pilnai sujungtų sluoksnių blokas surenka visus ankstesnių sluoksnių išgautus požymius ir remdamasis jais, priima galutinį sprendimą klasei priskirti vaizdą [20].




**5 pav.** Bendroji konvoliucinio neuroninio tinklo architektūros schema [21]

Tobulėjant CNN architektūroms, jos buvo papildytos keliais svarbiais projektavimo sprendimais, kurie leido apmokyti gilesnius ir tikslesnius tinklus. Vienas svarbiausių – liekamosios jungtys (angl. *residual connections*), leidžiančios signalui ne tik eiti per sluoksnį, bet ir jį apeiti, taip pridedant pradinę reikšmę prie sluoksnio išvesties. Toks sprendimas sumažina nykstančių gradientų problemą, su kuria susiduriama treniruojant gilius tinklus, ir leidžia kurti šimtus sluoksnių turinčias architektūras [22]. Kiti patobulinimai apima tankiąsias jungtis tarp sluoksnių, subalansuotą tinklo gylio ir pločio mastelio keitimą bei didesnių konvoliucinių branduolių naudojimą – šie principai pritaikyti tokiose architektūrose kaip ResNet [22], EfficientNet [23] ir ConvNeXt [24].

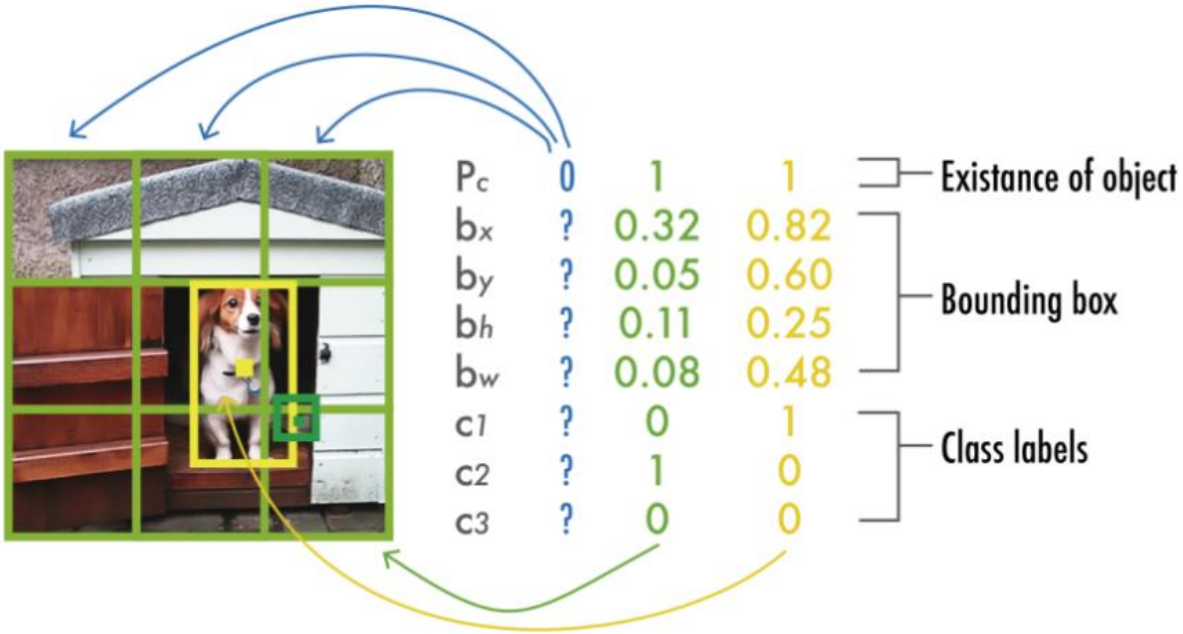
CNN architektūros plačiai taikomos vaizdų klasifikavimo, objektų aptikimo ir segmentavimo uždaviniuose. Jos naudojamos medicininių ir palydovinių vaizdų analizėje, robotikoje bei vaizdo stebėjimo sistemose. Vienas pagrindinių CNN pranašumų yra skaičiavimo efektyvumas ir santykinai nedidelis parametrų skaičius, nes tie patys filtrai taikomi skirtingose vaizdo vietose. Be to, šie tinklai geba atpažinti vietinius vaizdo požymius nepriklausomai nuo jų padėties, todėl gali veikti efektyviai net ir naudojant mažesnius duomenų rinkinius [20]. Tačiau, klasikinės CNN architektūros turi ir apribojimų. Konvoliucinė operacija yra lokali, todėl kiekviena išvesties reikšmė priklauso tik nuo nedidelės vaizdo srities. Dėl šios priežasties sudėtinga efektyviai nustatyti tolimas priklausomybes tarp skirtingų vaizdo dalių. Nors ši problema iš dalies sprendžiama didinant tinklo gylį, efektyvaus priėmimo lauko (angl. *effective receptive field*) apribojimai vis tiek išlieka [25]. Šiuos trūkumus siekia įveikti transformerių architektūros. Jose naudojamas dėmesio mechanizmas leidžia tiesiogiai modeliuoti ryšius tarp skirtingų vaizdo sričių nepriklausomai

nuo jų tarpusavio atstumo. Tačiau toks metodas paprastai reikalauja daugiau duomenų ir didesnių skaičiavimo resursų [26].

### 1.3.2. Vienos stadijos detektoriai

Vienos stadijos detektoriai – tai specializuota CNN architektūros atmaina, sukurta greitam objektų aptikimui realiu laiku. Pats populiariausias šios kategorijos detektorius yra YOLO (angl. *You Only Look Once*), plačiai taikomas kompiuterinės regos užduotyse. Skirtingai nuo dviejų stadijų metodų, tokių kaip Faster R-CNN [27], kurie pirmajame žingsnyje generuoja kandidatinius regionus, o antrajame juos klasifikuoja, YOLO atlieka aptikimą ir klasifikavimą vienu nuosekliu praėjimu per tinklą. Toks veikimo būdas reikšmingai padidina veikimo greitį, todėl YOLO architektūra yra labai tinkama realaus laiko aptikimo užduotims [28].

YOLO veikimas grindžiamas tinklelio principu, įvesties vaizdas padalijamas į S×S langelių tinklelį, ir kiekvienas langelis tiesiogiai prognozuoja ribojančius stačiakampius (angl. *bounding boxes*) bei klasių tikimybes objektams, kurių centras patenka į tą langelį. YOLO architektūrą sudaro trys pagrindiniai komponentai (žr. 6 pav.). Pagrindas (angl. *backbone*) yra konvoliucinis tinklas, iš įvesties vaizdo išgaunantis požymius – naujesnėse versijose tam dažniausiai naudojama CSPDarknet architektūra. Kaklas (angl. *neck*) sujungia skirtingo detalumo požymius iš įvairių pagrindo sluoksnių; čia paprastai taikoma FPN+PAN (angl. *Feature Pyramid Network + Path Aggregation Network*) struktūra, leidžianti vienu metu atsižvelgti tiek į smulkias detales, tiek į bendrą vaizdo kontekstą. Aptikimo galva (angl. *detection head*) iš sujungtų požymių prognozuoja galutines stačiakampių koordinates ir klasių tikimybes [28].




**6 pav.** YOLO šeimos detektoriaus architektūros schema [28]

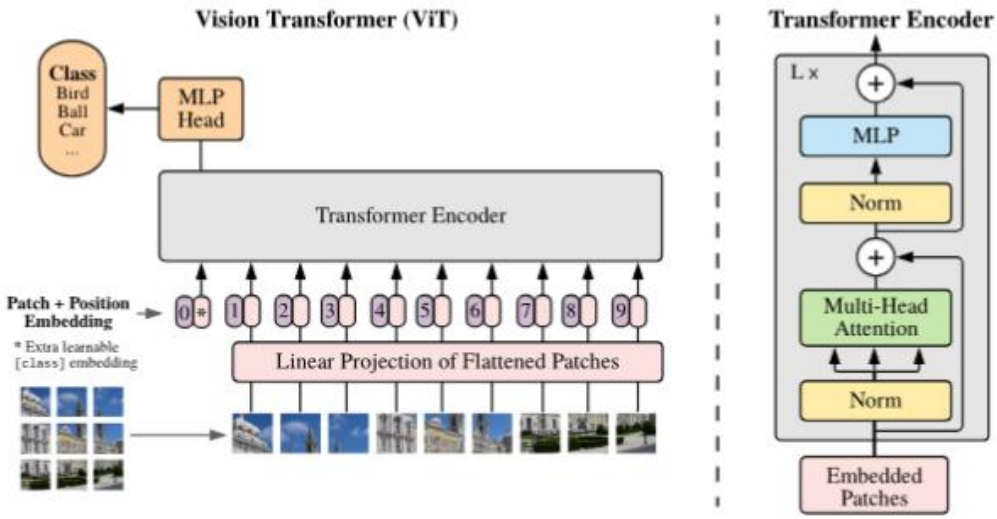
Pagrindinis vienos stadijos detektorių pranašumas – didelis veikimo greitis. Atliekant aptikimą ir klasifikavimą vienu praėjimu, skaičiavimo sąnaudos yra žymiai mažesnės nei dviejų stadijų metoduose, todėl vaizdus įmanoma apdoroti realiu laiku. Tokia architektūra

lanksti – ji gali veikti tiek savarankiškai, kai vienas modelis iš karto klasifikuoja aptiktus objektus, tiek kaip pirmoji grandis sudėtingesnėje sistemoje, kurioje detalesnį klasifikavimą atlieka atskiras specializuotas modelis.

Nepaisant greičio pranašumų, vienos stadijos detektoriai turi apribojimų. Dėl konvoliucinių tinklų lokalumo apribojimų, vienos stadijos modeliams sunkiau modeliuoti tolimus ryšius tarp skirtingų vaizdo dalių. Be to, vienu metu atliekamas aptikimas ir klasifikavimas gali sumažinti tikslumą scenose su tankiai išsidėsčiusiais arba stipriai persidengiančiais objektais. Tokiose situacijose dviejų stadijų metodai paprastai pasiekia geresnių rezultatų, todėl praktikoje pasirinkimas tarp vienos ir dviejų stadijų metodų dažnai priklauso nuo to, kas svarbiau: greitis ar tikslumas sudėtingose scenose [27].

### 1.3.3. Transformerių architektūros

Transformerių architektūra remiasi dėmesio mechanizmu, leidžiančiu modeliui dinamiškai įvertinti, kurios įvesties dalys yra svarbiausios kiekvienai išvesties prognozei. Skirtingai nuo konvoliucinių tinklų, kuriems būdingas lokalumo apribojimas, dėmesio mechanizmas leidžia tiesiogiai modeliuoti tolimas priklausomybes tarp bet kurių dviejų įvesties elementų, nepriklausomai nuo jų erdvinio atstumo [29]. Pagrindinė savidėmesio operacija veikia naudojant tris matricas – užklausų, raktų ir reikšmių. Kiekvienai įvesties pozicijai modelis apskaičiuoja jos panašumą su visomis kitomis pozicijomis, gautas reikšmes paverčia normalizuotais svoriais ir apskaičiuoja jų pagrindu sudarytą svorinę reikšmių sumą. Tokiu būdu tinklas vienu metu mato visą įvestį ir išmoksta, kurios jos dalys yra svarbiausios konkrečiai užduočiai. Daugiagalvinis savidėmesys (angl. *multi-head self-attention*) išplečia šią idėją – lygiagrečiai vykdomi keli dėmesio skaičiavimai skirtingose požymių poerdvėse, todėl modelis vienu metu gali užfiksuoti kelis skirtingus ryšių tipus tarp tų pačių elementų [29]. Vaizdo transformeris (angl. *Vision Transformer, ViT*) pritaiko šį principą kompiuterinės regos užduotims (žr. 7 pav.) [26]. Jo veikimas pagrįstas vaizdo skaidymu į fragmentus, kurie apdorojami panašiai kaip žodžiai natūralios kalbos modeliuose.




**7 pav.** Vaizdo transformerio architektūros schema [26]

Įvesties vaizdas padalijamas į fiksuoto dydžio fragmentus, kurie apdorojami kaip seka, analogiškai žodžiams natūralios kalbos modeliuose. Kiekvienas fragmentas paverčiamas įterpiniu, o jo pozicijai vaizde išsaugoti papildomai taikomas pozicinis kodavimas. Tuomet gauta seka perduodama transformerio koduotuvui, kuris informaciją apdoroja savidėmesio mechanizmo pagalba.

Transformerių architektūros kompiuterinėje regoje gali pasiekti, o kai kuriais atvejais ir pranokti, konvoliucinių neuroninių tinklų rezultatus. Vis dėlto jos turi ir tam tikrų trūkumų. Vienas pagrindinių – savidėmesio mechanizmo skaičiavimo sudėtingumas, kuris didėja kvadratiškai augant vaizdo fragmentų skaičiui. Dėl to didelės raiškos vaizdų apdorojimas reikalauja daug skaičiavimo resursų. Kita problema – transformeriai neturi CNN architektūroms būdingų induktyviųjų poslinkių, tokių kaip gebėjimas atpažinti vietines vaizdo struktūras ir išlaikyti objekto atpažinimą nepriklausomai nuo jo padėties vaizde. Dėl šios priežasties jų efektyviam apmokymui dažniausiai reikia didesnių duomenų rinkinių [26]. Siekiant sumažinti šiuos apribojimus, kuriami įvairūs transformerių variantai. Pavyzdžiui, objektų aptikimo modeliai RT-DETR [30] ir RF-DETR [31] tradicines inkarines dėžes pakeičia tiesioginiu objektų rinkinių prognozavimu. Tuo tarpu hierarchiniai transformeriai dėmesio skaičiavimą apriboja vietiniais langais, todėl sumažėja skaičiavimo sudėtingumas ir efektyviau apdorojami didelės raiškos vaizdai.

# 2. Projektinė dalis

## 2.1. Reikalavimų specifikacija

### 2.1.1. Projekto kūrimo pagrindas

Atliekamas tyrimas, kurio tikslas – pritaikyti konvoliucinių neuroninių tinklų (CNN) metodus siekiant automatizuoti pažeistų medžių lajų klasifikavimą. Šiuo tyrimu siekiama prisidėti prie ankstyvojo ligų nustatymo ir stebėsenos miškuose pažangos, sukurti patikimą ir tikslią sistemą, skirtą automatizuotam ligų aptikimui ir klasifikavimui miško ekosistemose. Sistemos kūrimo priežastys:

**Priežastis 1**. Brangus ir žmogiškiesiems resursams imlus miško medžių sveikatos ekspertinis įvertinimas reikalauja daug specialistų laiko ir pastangų.

**Priežastis 2**. Tyrimai vykdomi dabartiniais metodais nepadengia didelio ploto miškų – tik pasirinktas dalis.

### 2.1.2. Sistemos tikslai

**Tikslas 1.** Patikimai ir pigiau atlikti medžių ligų klasifikavimą.

**Pagrindimas:** Priežastys 1

**Tenkinimo kriterijus:** Sistema yra pakankamai tiksli, kad galėtų būti naudojama ankstyvam ligų nustatymui; sistema gali automatiškai nustatyti pažeistas medžių lajas ne didesne nei 10 % paklaida lyginant su dalykinės srities ekspertais.

**Tikslas 2.** Išplėsti ir pagreitinti miško medžių sveikatos vertinimo aprėptį ir procesą.

**Pagrindimas:** Priežastis 2

**Tenkinimo kriterijus:** Sistema gali būti įdiegta bet kur, aprėpti didelius miško plotus.

### 2.1.3. Apribojimai sprendimui

Apribojimas 1. Sistemos vartotojo sąsaja turi veikti naršyklėje naudojant skirtingus įrenginius.

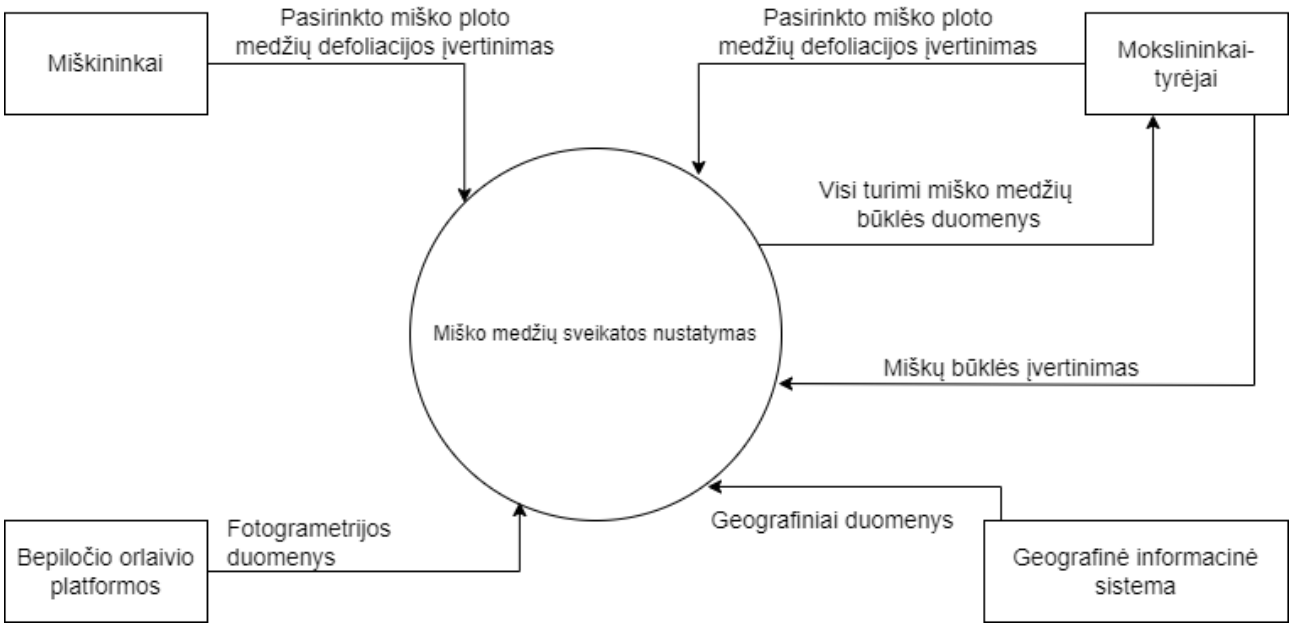
**Pagrindimas:** Sistemos veikimas turi būti lengvai prieinamas, veikti daugumos plačiosios visuomenės naudojamuose įrenginiuose.

**Tenkinimo kriterijus:** Sistema turi būti prieinama ir veikti pagrindinėse naršyklėse pvz. *Chrome*, *Firefox*, *Safari, Edge*.

### 2.1.4. Esama padėtis

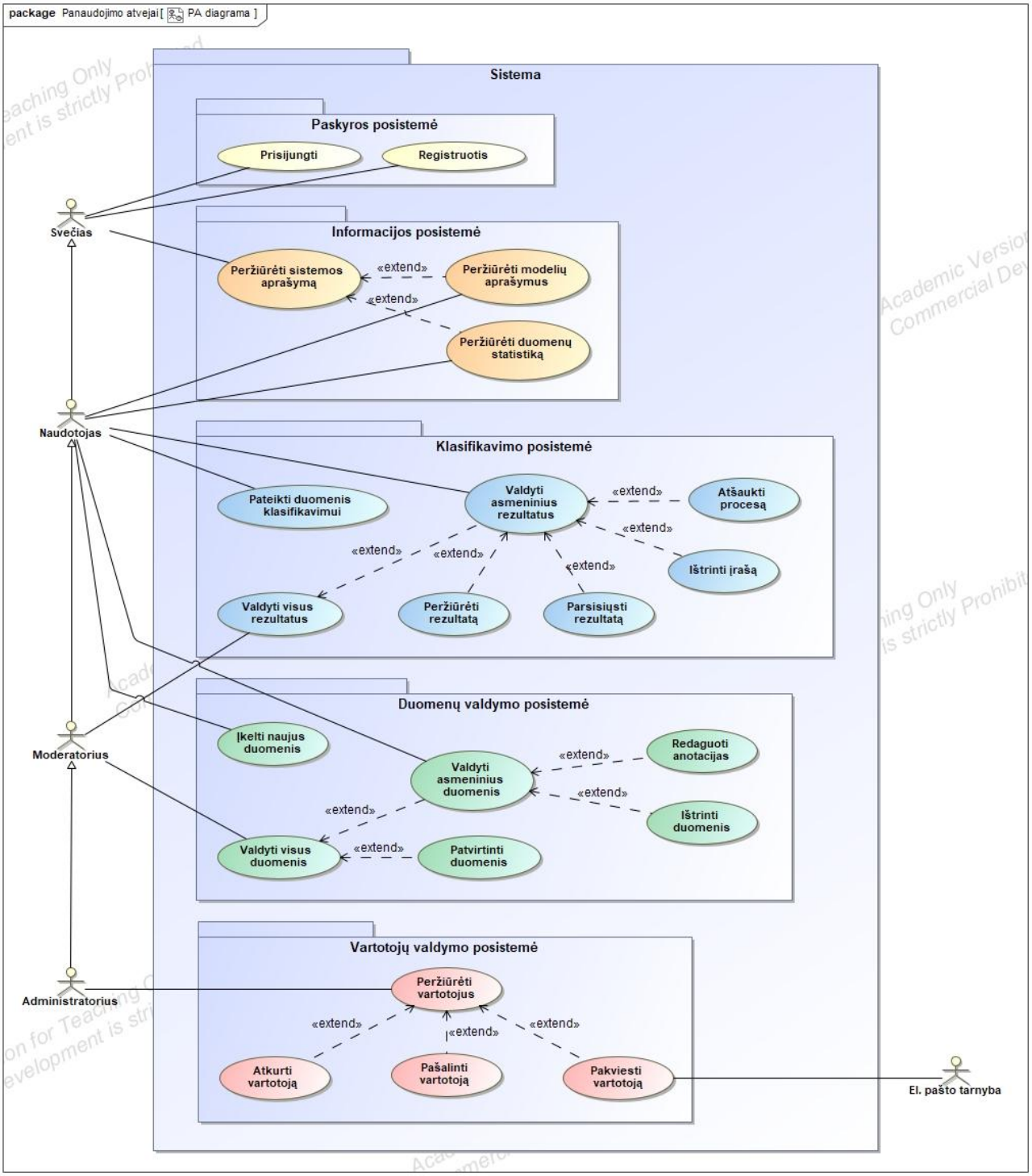
Šiuo metu miško medžių lapų defoliacijos stebėsena ir įvertinimas yra vykdomi lauko tyrimo būdu. Miško inžinieriai ir tyrėjai vykdo ekspedicijas, kurių metu tikrina pasirinktus miškų plotus, kur vizualiai vertina medžių defoliacijos lygį. Ekspertai stebi ir įvertina medžių vainikus, lajas, ieškodami pažeidimų ar lapų praradimo požymių ir pagal iš anksto nustatytus kriterijus klasifikuoja defoliacijos lygį – vėliau duomenis susistemina.

### 2.1.5. Veiklos kontekstas




**8 pav.** Veiklos kontekstas

### 2.1.6. Sistemos panaudojimo atvejai




**9 pav.** Panaudojimo atvejų diagrama

**1 lentelė** PA „Prisijungti“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 1 | Prisijungti |
| Tikslas | Prisijungti prie paskyros |
| Dalyviai | Naudotojas |
| Priklausomybės | – |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką |

| | |
|---|---|
| Prieš-sąlygos | Naudotojas turi aktyvią paskyrą |
| Sužadinimo sąlyga | Naudotojas pasirenka prisijungti |
| Po-sąlyga | Naudotojas prijungiamas prie turimos paskyros |
| Pagrindinis scenarijus | Naudotojas inicijuoja prisijungimą prie paskyros, sistema paprašo įvesti prisijungimo vardą ir slaptažodį, sėkmingai prisijungus pateikiamas sistemos funkcionalumus pagal turimas teises |
| Alternatyvūs scenarijai | Jei naudotojas pateikia neteisingus prisijungimo duomenis, sistema prisijungimą atmeta ir pateikia klaidos pranešimą |

**2 lentelė** PA „Registruotis“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 2 | Registruotis |
| Tikslas | Užsiregistruoti sistemoje pagal gautą pakvietimą |
| Dalyviai | Svečias |
| Priklausomybės | – |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką |
| Prieš-sąlygos | Naudotojas turi galiojantį, nepanaudotą pakvietimo žetoną |
| Sužadinimo sąlyga | Naudotojas atidaro registracijos nuorodą su žetonu |
| Po-sąlyga | Naudotojas užregistruotas sistemoje ir gali prisijungti |
| Pagrindinis scenarijus | Naudotojas atidaro registracijos nuorodą, o sistema siunčia pakvietimo žetoną patvirtinimui ir pateikia rezultatą naudotojui |
| Alternatyvūs scenarijai | Jei registracijos žetonas negalioja, formos duomenys neteisingi arba slapyvardis jau užimtas sistema registracijos neįvykdo ir pateikia klaidos pranešimą |

**3 lentelė** PA „Peržiūrėti sistemos aprašymą“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 3 | Peržiūrėti sistemos aprašymą |
| Tikslas | Peržiūrėti bendrą informaciją apie sistemą |
| Dalyviai | Svečias, naudotojas |
| Priklausomybės | Peržiūrėti modelių aprašymus, Peržiūrėti duomenų statistiką |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką |
| Prieš-sąlygos | – |
| Sužadinimo sąlyga | Naudotojas inicijuoja sistemos informacijos peržiūrą |
| Po-sąlyga | Naudotojas mato sistemos aprašymą |
| Pagrindinis scenarijus | Naudotojas inicijuoja sistemos informacijos peržiūrą, o sistema užklausia bendrą informaciją ir pateikia rezultatą naudotojui |
| Alternatyvūs scenarijai | – |

**4 lentelė** PA „Peržiūrėti modelių aprašymus“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 4 | Peržiūrėti modelių aprašymus |
| Tikslas | Peržiūrėti sistemoje naudojamų klasifikavimo modelių informaciją |
| Dalyviai | Naudotojas |
| Priklausomybės | – |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką |

| | |
|---|---|
| Prieš-sąlygos | Naudotojas yra prisijungęs prie sistemos |
| Sužadinimo sąlyga | Naudotojas pasirenka „Modeliai" skirtuką informacijos puslapyje |
| Po-sąlyga | Naudotojas mato modelių aprašymus |
| Pagrindinis scenarijus | Naudotojas pasirenka „Modeliai" skirtuką, o sistema užklausia modelių informaciją ir pateikia rezultatą naudotojui |
| Alternatyvūs scenarijai | – |

**5 lentelė** PA „Peržiūrėti duomenų statistiką“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 5 | Peržiūrėti duomenų statistiką |
| Tikslas | Peržiūrėti sistemos surinktų duomenų statistiką |
| Dalyviai | Naudotojas |
| Priklausomybės | – |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką |
| Prieš-sąlygos | Naudotojas yra prisijungęs prie sistemos |
| Sužadinimo sąlyga | Naudotojas pasirenka „Statistika" skirtuką informacijos puslapyje |
| Po-sąlyga | Naudotojas mato duomenų statistiką |
| Pagrindinis scenarijus | Naudotojas pasirenka „Statistika" skirtuką, o sistema užklausia statistikos duomenis ir pateikia rezultatą naudotojui |
| Alternatyvūs scenarijai | – |

**6 lentelė** PA „Pateikti duomenis klasifikavimui“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 6 | Pateikti duomenis klasifikavimui |
| Tikslas | Pateikti nuotrauką ar vaizdo įrašą automatiniam klasifikavimui |
| Dalyviai | Naudotojas |
| Priklausomybės | – |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką |
| Prieš-sąlygos | Naudotojas yra prisijungęs prie sistemos |
| Sužadinimo sąlyga | Naudotojas inicijuoja failo pateikimą klasifikavimui |
| Po-sąlyga | Naudotojas mato klasifikavimo rezultatą su anotacijomis |
| Pagrindinis scenarijus | Naudotojas inicijuoja failo pateikimą klasifikavimui, o sistema pateikia klasifikavimo pateikties formą ir pateikia rezultatą naudotojui |
| Alternatyvūs scenarijai | – |

**7 lentelė** PA „Valdyti asmeninius rezultatus“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 7 | Valdyti asmeninius rezultatus |
| Tikslas | Valdyti savo klasifikavimo rezultatus |
| Dalyviai | Naudotojas |
| Priklausomybės | Išplečiantys: Atšaukti procesą, Peržiūrėti rezultatą, Parsisiųsti rezultatą, Ištrinti įrašą; Specializuoti: Valdyti visus rezultatus |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką |
| Prieš-sąlygos | Naudotojas yra prisijungęs prie sistemos |
| Sužadinimo sąlyga | Naudotojas inicijuoja rezultatų valdymą |

| | |
|---|---|
| Po-sąlyga | Naudotojas mato savo klasifikavimo rezultatų sąrašą |
| Pagrindinis scenarijus | Naudotojas inicijuoja rezultatų valdymą, o sistema užklausia naudotojo klasifikavimo darbų sąrašą ir pateikia rezultatą naudotojui |
| Alternatyvūs scenarijai | – |

**8 lentelė** PA „Atšaukti procesą“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 8 | Atšaukti procesą |
| Tikslas | Atšaukti vykdomą klasifikavimo procesą |
| Dalyviai | Naudotojas |
| Priklausomybės | – |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką |
| Prieš-sąlygos | Naudotojas turi vykdomą (laukiančią arba apdorojamą) klasifikavimo užduotį |
| Sužadinimo sąlyga | Naudotojas pasirenka proceso atšaukimą |
| Po-sąlyga | Klasifikavimo procesas atšauktas, rezultato būsena – „Atšaukta" |
| Pagrindinis scenarijus | Naudotojas pasirenka proceso atšaukimą, o sistema siunčia atšaukimo užklausą ir pateikia rezultatą naudotojui |
| Alternatyvūs scenarijai | – |

**9 lentelė** PA „Valdyti visus rezultatus“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 9 | Valdyti visus rezultatus |
| Tikslas | Valdyti visų naudotojų klasifikavimo rezultatus |
| Dalyviai | Moderatorius |
| Priklausomybės | Atšaukti procesą, Peržiūrėti rezultatą, Parsisiųsti rezultatą, Ištrinti įrašą, Valdyti asmeninius rezultatus |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką |
| Prieš-sąlygos | Naudotojas yra prisijungęs prie sistemos |
| Sužadinimo sąlyga | Moderatorius pasirenka „Visi klasifikavimo darbai" skirtuką |
| Po-sąlyga | Moderatorius mato visų naudotojų klasifikavimo rezultatų sąrašą |
| Pagrindinis scenarijus | Naudotojas moderatorius pasirenka „Visi klasifikavimo darbai" skirtuką, o sistema užklausia visų naudotojų klasifikavimo darbų sąrašą ir pateikia rezultatą naudotojui |
| Alternatyvūs scenarijai | – |

**10 lentelė** PA „Peržiūrėti rezultatą“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 10 | Peržiūrėti rezultatą |
| Tikslas | Peržiūrėti klasifikavimo rezultato detalią informaciją |
| Dalyviai | Naudotojas |
| Priklausomybės | – |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką |
| Prieš-sąlygos | Klasifikavimo darbas yra užbaigtas |
| Sužadinimo sąlyga | Naudotojas pasirenka rezultatą peržiūrai |
| Po-sąlyga | Naudotojas mato klasifikavimo rezultato detalią informaciją |

| | |
|---|---|
| Pagrindinis scenarijus | Naudotojas pasirenka rezultatą peržiūrai, o sistema užklausia rezultato informaciją ir failą ir pateikia rezultatą naudotojui |
| Alternatyvūs scenarijai | – |

**11 lentelė** PA „Parsisiųsti rezultatą“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 11 | Parsisiųsti rezultatą |
| Tikslas | Parsisiųsti klasifikavimo rezultato failą |
| Dalyviai | Naudotojas |
| Priklausomybės | – |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką |
| Prieš-sąlygos | Klasifikavimo darbas yra užbaigtas |
| Sužadinimo sąlyga | Naudotojas pasirenka rezultato parsisiuntimą |
| Po-sąlyga | Naudotojas parsisiuntė anotuotą klasifikavimo rezultato failą |
| Pagrindinis scenarijus | Naudotojas pasirenka rezultato parsisiuntimą, o sistema užklausia rezultato failą ir pateikia rezultatą naudotojui |
| Alternatyvūs scenarijai | – |

**12 lentelė** PA „Ištrinti įrašą“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 12 | Ištrinti įrašą |
| Tikslas | Ištrinti klasifikavimo rezultato įrašą |
| Dalyviai | Naudotojas |
| Priklausomybės | – |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką |
| Prieš-sąlygos | Klasifikavimo darbas yra užbaigtas, nepavykęs arba atšauktas |
| Sužadinimo sąlyga | Naudotojas pasirenka rezultato trynimą |
| Po-sąlyga | Klasifikavimo rezultatas ištrintas iš sistemos |
| Pagrindinis scenarijus | Naudotojas pasirenka rezultato trynimą, o sistema pateikia patvirtinimo prašymą ir pateikia rezultatą naudotojui |
| Alternatyvūs scenarijai | – |

**13 lentelė** PA „Įkelti naujus duomenis“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 13 | Įkelti naujus duomenis |
| Tikslas | Įkelti naują nuotrauką su anotacijomis į sistemą |
| Dalyviai | Naudotojas |
| Priklausomybės | – |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką |
| Prieš-sąlygos | Naudotojas yra prisijungęs prie sistemos |
| Sužadinimo sąlyga | Naudotojas inicijuoja duomenų įkėlimą |
| Po-sąlyga | Nuotrauka ir anotacijos išsaugotos sistemoje. Moderatoriaus/administratoriaus duomenys automatiškai patvirtinami |
| Pagrindinis scenarijus | Naudotojas inicijuoja duomenų įkėlimą, o sistema pateikia duomenų įkėlimo puslapį ir pateikia rezultatą naudotojui |

| | |
|---|---|
| Alternatyvūs scenarijai | Jei įkeltas failas yra netinkamo formato ar dydžio arba pateikti duomenys nekorektiški, sistema įkėlimo nepriima ir pateikia klaidos pranešimą |

**14 lentelė** PA „Valdyti asmeninius duomenis“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 14 | Valdyti asmeninius duomenis |
| Tikslas | Valdyti savo įkeltus duomenis |
| Dalyviai | Naudotojas |
| Priklausomybės | Redaguoti anotacijas, Ištrinti duomenis, Valdyti visus duomenis |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką |
| Prieš-sąlygos | Naudotojas yra prisijungęs prie sistemos |
| Sužadinimo sąlyga | Naudotojas inicijuoja duomenų valdymą |
| Po-sąlyga | Naudotojas mato savo įkeltų duomenų sąrašą |
| Pagrindinis scenarijus | Naudotojas inicijuoja duomenų valdymą, o sistema užklausia duomenų sąrašą ir pateikia rezultatą naudotojui |
| Alternatyvūs scenarijai | – |

**15 lentelė** PA „Valdyti visus duomenis“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 15 | Valdyti visus duomenis |
| Tikslas | Valdyti visų naudotojų įkeltus duomenis |
| Dalyviai | Moderatorius |
| Priklausomybės | Patvirtinti duomenis, Redaguoti anotacijas, Ištrinti duomenis, Valdyti asmeninius duomenis. |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką. |
| Prieš-sąlygos | Naudotojas yra moderatorius arba administratorius. |
| Sužadinimo sąlyga | Moderatorius pasirenka „Visos nuotraukos" skirtuką. |
| Po-sąlyga | Moderatorius mato visų naudotojų duomenų sąrašą. |
| Pagrindinis scenarijus | Naudotojas moderatorius pasirenka „Visos nuotraukos" skirtuką, o sistema užklausia visų naudotojų duomenų sąrašą ir pateikia rezultatą naudotojui. |
| Alternatyvūs scenarijai | – |

**16 lentelė** PA „Patvirtinti duomenis“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 16 | Patvirtinti duomenis |
| Tikslas | Patvirtinti naudotojo įkeltus duomenis |
| Dalyviai | Moderatorius |
| Priklausomybės | – |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką |
| Prieš-sąlygos | Nuotrauka yra nepatvirtinta |
| Sužadinimo sąlyga | Moderatorius pasirenka nuotraukos patvirtinimą |
| Po-sąlyga | Nuotrauka patvirtinta, naudotojas nebegali jos redaguoti |
| Pagrindinis scenarijus | Naudotojas moderatorius pasirenka nuotraukos patvirtinimą, o sistema siunčia patvirtinimo užklausą ir pateikia rezultatą naudotojui |
| Alternatyvūs scenarijai | – |

**17 lentelė** PA „Redaguoti anotacijas“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 17 | Redaguoti anotacijas |
| Tikslas | Redaguoti nuotraukos anotacijas |
| Dalyviai | Naudotojas, moderatorius |
| Priklausomybės | – |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką |
| Prieš-sąlygos | Naudotojas turi teisę redaguoti nuotrauką |
| Sužadinimo sąlyga | Naudotojas pasirenka nuotraukos redagavimą |
| Po-sąlyga | Anotacijos atnaujintos duomenų bazėje |
| Pagrindinis scenarijus | Naudotojas pasirenka nuotraukos redagavimą, o sistema užklausia nuotrauką su anotacijomis ir pateikia rezultatą naudotojui |
| Alternatyvūs scenarijai | Jei naudotojas neturi teisės redaguoti (pvz., nuotrauka patvirtinta arba nepriklauso jam), sistema redagavimo neleidžia ir pateikia klaidos pranešimą |

**18 lentelė** PA „Ištrinti duomenis“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 18 | Ištrinti duomenis |
| Tikslas | Ištrinti įkeltus duomenis iš sistemos |
| Dalyviai | Naudotojas, moderatorius |
| Priklausomybės | – |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką |
| Prieš-sąlygos | Nuotrauka egzistuoja sistemoje |
| Sužadinimo sąlyga | Naudotojas pasirenka duomenų trynimą |
| Po-sąlyga | Nuotrauka ištrinta arba paslėpta nuo naudotojo |
| Pagrindinis scenarijus | Naudotojas pasirenka nuotraukos trynimą, o sistema pateikia patvirtinimo prašymą ir pateikia rezultatą naudotojui |
| Alternatyvūs scenarijai | – |

**19 lentelė** PA „Peržiūrėti naudotojus“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 19 | Peržiūrėti naudotojus |
| Tikslas | Peržiūrėti registruotų naudotojų sąrašą |
| Dalyviai | Administratorius |
| Priklausomybės | Pakviesti naudotoją, Pašalinti naudotoją, Atkurti naudotoją |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką |
| Prieš-sąlygos | Naudotojas yra administratorius |
| Sužadinimo sąlyga | Administratorius inicijuoja naudotojų peržiūrą |
| Po-sąlyga | Administratorius mato naudotojų sąrašą |
| Pagrindinis scenarijus | Naudotojas administratorius inicijuoja naudotojų peržiūrą, o sistema užklausia naudotojų sąrašą ir pateikia rezultatą naudotojui |
| Alternatyvūs scenarijai | – |

**20 lentelė** PA „Pakviesti naudotoją“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 20 | Pakviesti naudotoją |

| | |
|---|---|
| Tikslas | Pakviesti naują naudotoją registruotis sistemoje. |
| Dalyviai | Administratorius. |
| Priklausomybės | – |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką. |
| Prieš-sąlygos | Naudotojas yra administratorius. |
| Sužadinimo sąlyga | Administratorius inicijuoja naudotojo pakvietimą. |
| Po-sąlyga | Pakvietimas sukurtas, el. laiškas išsiųstas nurodytu adresu. |
| Pagrindinis scenarijus | Naudotojas administratorius inicijuoja naudotojo pakvietimą, o sistema pateikia pakvietimo formą ir pateikia rezultatą naudotojui. |
| Alternatyvūs scenarijai | Jei pakvietimo formos duomenys neteisingi arba naudotojas su nurodytu el. paštu jau egzistuoja, sistema pakvietimo nesukuria ir administratoriui pateikia klaidos pranešimą. |

**21 lentelė** PA „Pašalinti naudotoją“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 21 | Pašalinti naudotoją |
| Tikslas | Pašalinti naudotoją iš sistemos |
| Dalyviai | Administratorius |
| Priklausomybės | – |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką |
| Prieš-sąlygos | Naudotojas yra administratorius, trinamas naudotojas nėra pats administratorius |
| Sužadinimo sąlyga | Administratorius pasirenka naudotojo pašalinimą |
| Po-sąlyga | Naudotojas pažymėtas kaip pašalintas, nebegali prisijungti |
| Pagrindinis scenarijus | Naudotojas administratorius pasirenka naudotojo pašalinimą, o sistema pateikia patvirtinimo prašymą ir pateikia rezultatą naudotojui |
| Alternatyvūs scenarijai | – |

**22 lentelė** PA „Atkurti naudotoją“ specifikacija

| | |
|---|---|
| Panaudojimo atvejis 22 | Atkurti naudotoją |
| Tikslas | Atkurti pašalintą naudotoją. |
| Dalyviai | Administratorius. |
| Priklausomybės | – |
| Nefunkciniai reikalavimai | Sistema turi užtikrinti stabilų veikimą ir pateikti rezultatą per priimtiną laiką. |
| Prieš-sąlygos | Naudotojas yra administratorius, atkuriamas naudotojas yra pašalintas. |
| Sužadinimo sąlyga | Administratorius pasirenka naudotojo atkūrimą. |
| Po-sąlyga | Naudotojas atkurtas, gali vėl prisijungti prie sistemos. |
| Pagrindinis scenarijus | Naudotojas administratorius pasirenka naudotojo atkūrimą, o sistema siunčia atkūrimo užklausą ir pateikia rezultatą naudotojui. |
| Alternatyvūs scenarijai | – |

## 2.1.7. Funkciniai reikalavimai ir reikalavimai duomenims

### 2.1.7.1. Funkciniai reikalavimai

**23 lentelė** Klasifikavimo algoritmo reikalavimas

| | | | | | |
|---|---|---|---|---|---|
| **Reikalavimas #:** | **FR-1** | **Reikalavimo tipas:** | **V9** | **PA:** | **1** |
| **Aprašymas:** | Sistema turi naudoti konvoliucinius neuroninius tinklus medžių nuotraukų klasifikavimui atlikti. | | | | |
| **Pagrindimas:** | Konvoliuciniai neuroniniai tinklai yra populiariausia mašininio mokymosi kategorija nuotraukų klasifikavimui atlikti. | | | | |
| **Šaltinis:** | Studentas | | | | |
| **Atitikimo kriterijus:** | Sistemos klasifikavimo algoritmas realizuotas panaudojus konvoliucinius neuroninius tinklus. | | | | |
| **Užsakovo tenkinimas:** | 3 | **Užsakovo netenkinimas:** | 5 | | |
| **Prioritetas:** | Aukštas | **Konfliktai:** | Nėra | | |
| **Istorija:** | Sukurtas: 2024-03-04 | | | | |

**24 lentelė** Duomenų apdorojimo reikalavimas

| | | | | | |
|---|---|---|---|---|---|
| **Reikalavimas #:** | **FR-2** | **Reikalavimo tipas:** | **V9** | **PA:** | **1** |
| **Aprašymas:** | Sistema turi atlikti pirminį duomenų apdorojimą pvz. dydžio sumažinimą; filtravimą; šviesos, kontrasto koregavimą siekiant pagerinti klasifikavimo tikslumą. | | | | |
| **Pagrindimas:** | Siekiama sukurti kuo tikslesnį ir patikimesnį klasifikavimo algoritmą. | | | | |
| **Šaltinis:** | Studentas | | | | |
| **Atitikimo kriterijus:** | Sistemos klasifikavimo algoritmui įgyvendintas duomenų apdorojimas, standartizavimas, filtravimas ar kitos apmokymo kokybę gerinančios priemonės. | | | | |
| **Užsakovo tenkinimas:** | 4 | **Užsakovo netenkinimas:** | 5 | | |
| **Prioritetas:** | Aukštas | **Konfliktai:** | Nėra | | |
| **Istorija:** | Sukurtas: 2024-03-04 | | | | |

**25 lentelė** Duomenų importavimo reikalavimas

| | | | | | |
|---|---|---|---|---|---|
| **Reikalavimas #:** | **FR-3** | **Reikalavimo tipas:** | **V9** | **PA:** | **1,3,4,5** |
| **Aprašymas:** | Sistema turi turėti galimybę importuoti duomenis-miško nuotraukas iš skirtingų šaltinių, failų. | | | | |
| **Pagrindimas:** | Surinkti kuo didesnį duomenų rinkinį klasifikavimo algoritmo apmokymui. | | | | |
| **Šaltinis:** | Studentas | | | | |
| **Atitikimo kriterijus:** | Sistema gali importuoti ir valdyti skirtingo tipo miško medžių nuotraukas. | | | | |
| **Užsakovo tenkinimas:** | 3 | **Užsakovo netenkinimas:** | 4 | | |
| **Prioritetas:** | Vidutinis | **Konfliktai:** | Nėra | | |
| **Istorija:** | Sukurtas: 2024-03-10 | | | | |

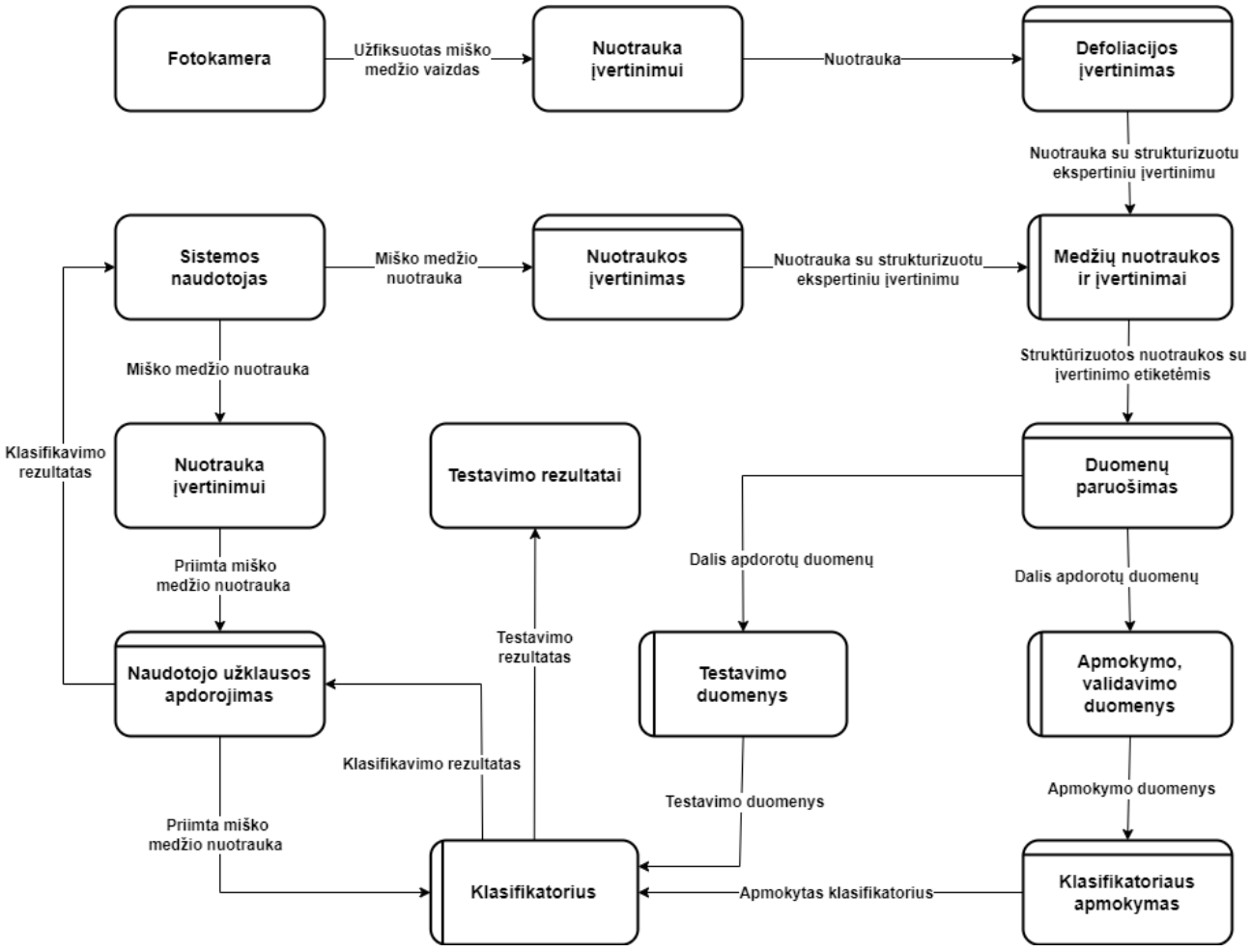
**26 lentelė** Modelio informacijos pateikimo reikalavimas

| | | | | | |
|---|---|---|---|---|---|
| **Reikalavimas #:** | **FR-4** | **Reikalavimo tipas:** | **V9** | **PA:** | **2** |
| **Aprašymas:** | Sistema turi pateikti informaciją apie klasifikavimo modelio duomenis. | | | | |

| | | | |
|---|---|---|---|
| **Pagrindimas:** | Naudotojui pateikti informaciją apie klasifikavimo modelio apmokymo duomenis, kad jis galėtų įvertinti modelio patikimumą. | | |
| **Šaltinis:** | Studentas | | |
| **Atitikimo kriterijus:** | Sistema pateikia svarbią susistemintą informaciją apie apmokymo duomenis. | | |
| **Užsakovo tenkinimas:** | 2 | **Užsakovo netenkinimas:** | 1 |
| **Prioritetas:** | Žemas | **Konfliktai:** | Nėra |
| **Istorija:** | Sukurtas: 2024-03-10 | | |

### 2.1.7.2. Reikalavimai duomenims

Žemiau pateikiamas pradinis duomenų modelis (10 pav.).




**10 pav.** Duomenų modelis

## 2.2. Nefunkciniai reikalavimai

### 2.2.1. Reikalavimai sistemos išvaizdai

**27 lentelė** Naudotojo sąsajos intuityvumo reikalavimas

| | | | | | |
|---|---|---|---|---|---|
| **Reikalavimas #:** | **NR-1** | **Reikalavimo tipas:** | **V10** | **PA:** | **visi** |

| | | | |
|---|---|---|---|
| **Aprašymas:** | Sistemos sąsaja turi būti intuityvi ir lengvai suprantama visiems vartotojams nepriklausomai nuo IT patirties. | | |
| **Pagrindimas:** | Kad vidutinis naudotojas, nepriklausomai nuo amžiaus ar išsilavinimo, greitai perprastų sistemos veikimą. | | |
| **Šaltinis:** | Studentas | | |
| **Atitikimo kriterijus:** | Sistemos vartotojo sąsaja yra intuityvi, pirminiai naudotojai patvirtina, kad lengva orientuotis sistemoje, ji yra lengvai suprantama. | | |
| **Užsakovo tenkinimas:** | 1 | **Užsakovo netenkinimas:** | 1 |
| **Prioritetas:** | Žemas | **Konfliktai:** | Nėra |
| **Istorija:** | Sukurtas: 2024-03-04 | | |

**28 lentelė** Sąsajos vientisumo reikalavimas

| | | | | | |
|---|---|---|---|---|---|
| **Reikalavimas #:** | **NR-2** | **Reikalavimo tipas:** | **V10** | **PA:** | **visi** |
| **Aprašymas:** | Turi būti užtikrinta nuosekli išvaizda visoje sistemos naudotojo sąsajoje. | | | | |
| **Pagrindimas:** | Kad vartotojai jaustų aiškumą ir vientisumą, aiškiai suprastų kaip atrodo atskiri elementai, juos būtų lengva identifikuoti. | | | | |
| **Šaltinis:** | Studentas | | | | |
| **Atitikimo kriterijus:** | Sukurtos sistemos elementai yra standartizuoti, atsikartoja per visą naudotojo sąsają. | | | | |
| **Užsakovo tenkinimas:** | 1 | **Užsakovo netenkinimas:** | 1 | | |
| **Prioritetas:** | Žemas | **Konfliktai:** | Nėra | | |
| **Istorija:** | Sukurtas: 2024-03-04 | | | | |

**29 lentelė** Klaidų valdymo reikalavimas

| | | | | | |
|---|---|---|---|---|---|
| **Reikalavimas #:** | **NR-3** | **Reikalavimo tipas:** | **V10** | **PA:** | **visi** |
| **Aprašymas:** | Sistema turi galėti valdyti naudotojo klaidas, pateikti grįžtamąsias žinutes apie neteisingai naudojamą sistemą. | | | | |
| **Pagrindimas:** | Kad naudotojams būtų aišku kaip naudotis sistema, jei ją naudoja ne taip kaip numatyta, įveda neteisingus duomenis. | | | | |
| **Šaltinis:** | Studentas | | | | |
| **Atitikimo kriterijus:** | Naudotojui pateikus netinkamus duomenis sistema pateikia naudingą klaidos pranešimą. | | | | |
| **Užsakovo tenkinimas:** | 2 | **Užsakovo netenkinimas:** | 4 | | |
| **Prioritetas:** | Žemas | **Konfliktai:** | Nėra | | |
| **Istorija:** | Sukurtas: 2024-03-04 | | | | |

**30 lentelė** Laukimo pranešimų reikalavimas

| | | | | | |
|---|---|---|---|---|---|
| **Reikalavimas #:** | **NR-4** | **Reikalavimo tipas:** | **V10** | **PA:** | **visi** |
| **Aprašymas:** | Sistema turi pateikti laukimo pranešimus, kai numatoma, kad naudotojui reikės palaukti t.y. duomenų parsisiuntimo, perdavimo serveriui, klasifikavimo metu. | | | | |
| **Pagrindimas:** | Naudotojai yra nekantrūs. | | | | |
| **Šaltinis:** | Studentas | | | | |
| **Atitikimo kriterijus:** | Naudotojui pateikus nuotrauką įvertinimui, pateikiamas pranešimas, kad duomenys apdorojami. | | | | |

| | | | |
|---|---|---|---|
| **Užsakovo tenkinimas:** | 3 | **Užsakovo netenkinimas:** | 4 |
| **Prioritetas:** | Žemas | **Konfliktai:** | Nėra |
| **Istorija:** | Sukurtas: 2024-03-04 | | |

### 2.2.2. Reikalavimai panaudojamumui

**31 lentelė** Naudojimosi paprastumo reikalavimas

| | | | | | |
|---|---|---|---|---|---|
| **Reikalavimas #:** | **NR-5** | **Reikalavimo tipas:** | **V11** | **PA:** | **visi** |
| **Aprašymas:** | Sistemos navigacija, valdymas turi būti labai paprastas pvz. nuoseklus tiesinis valdymas. | | | | |
| **Pagrindimas:** | Kad visi naudotojai galėtų lengvai perprasti, naudotis sistema. | | | | |
| **Šaltinis:** | Studentas | | | | |
| **Atitikimo kriterijus:** | Testuotojai, pirminiai naudotojai turėtų lengvai (per 10 minučių) perprasti sistemos veikimą, jos funkcijas. | | | | |
| **Užsakovo tenkinimas:** | 2 | **Užsakovo netenkinimas:** | 2 | | |
| **Prioritetas:** | Žemas | **Konfliktai:** | Nėra | | |
| **Istorija:** | Sukurtas: 2024-03-04 | | | | |

**32 lentelė** Kalbos reikalavimas

| | | | | | |
|---|---|---|---|---|---|
| **Reikalavimas #:** | **NR-6** | **Reikalavimo tipas:** | **V11** | **PA:** | **visi** |
| **Aprašymas:** | Sistema turi būti sukonfigūruota lietuvių kalbai. | | | | |
| **Pagrindimas:** | Sistemos tikslinė auditorija yra lietuviai. | | | | |
| **Šaltinis:** | Studentas | | | | |
| **Atitikimo kriterijus:** | Tekstas, navigacija sistemoje pateikta lietuvių kalba. | | | | |
| **Užsakovo tenkinimas:** | 2 | **Užsakovo netenkinimas:** | 1 | | |
| **Prioritetas:** | Žemas | **Konfliktai:** | Nėra | | |
| **Istorija:** | Sukurtas: 2024-03-04 | | | | |

### 2.2.3. Reikalavimai vykdymo charakteristikoms

**33 lentelė** Atsako greičio reikalavimas

| | | | | | |
|---|---|---|---|---|---|
| **Reikalavimas #:** | **NR-7** | **Reikalavimo tipas:** | **V12** | **PA:** | **visi** |
| **Aprašymas:** | Sistema turi būti interaktyvi, duoti atsaką vidutiniškai per mažiau nei sekundę (išskyrus duomenų parsisiuntimą, perdavimą serveriui, klasifikavimą). | | | | |
| **Pagrindimas:** | Vartotojai yra reiklūs, siekiama pristatyti kokybišką produktą. | | | | |
| **Šaltinis:** | Studentas | | | | |
| **Atitikimo kriterijus:** | Naviguojant, naudojant sistemos funkcijas sistema veikia greitai, vartotojui nereikia laukti funkcijų įvykdymo. | | | | |
| **Užsakovo tenkinimas:** | 3 | **Užsakovo netenkinimas:** | 4 | | |
| **Prioritetas:** | Žemas | **Konfliktai:** | Nėra | | |
| **Istorija:** | Sukurtas: 2024-03-04 | | | | |

### 2.2.4. Reikalavimai veikimo sąlygoms

**34 lentelė** Naršyklių suderinamumo reikalavimas

| | | | | | |
|---|---|---|---|---|---|
| **Reikalavimas #:** | **NR-8** | **Reikalavimo tipas:** | **V13** | **PA:** | **visi** |
| **Aprašymas:** | Sistema turi būti prieinama ir veikti pagrindinėse naršyklėse pvz. *Chrome, Firefox, Safari, Edge*. | | | | |
| **Pagrindimas:** | Sistemos veikimas turi būti lengvai prieinamas, veikti daugumos plačiosios visuomenės naudojamuose įrenginiuose. | | | | |
| **Šaltinis:** | Studentas | | | | |
| **Atitikimo kriterijus:** | Sistema ištestuota ir tinkamai veikia *Chrome, Firefox, Safari, Edge* naršyklėse. | | | | |
| **Užsakovo tenkinimas:** | 3 | **Užsakovo netenkinimas:** | 4 | | |
| **Prioritetas:** | Žemas | **Konfliktai:** | Nėra | | |
| **Istorija:** | Sukurtas: 2024-03-04 | | | | |

### 2.2.5. Reikalavimai sistemos priežiūrai

**35 lentelė** Modulinės architektūros reikalavimas

| | | | | | |
|---|---|---|---|---|---|
| **Reikalavimas #:** | **NR-9** | **Reikalavimo tipas:** | **V14** | **PA:** | **visi** |
| **Aprašymas:** | Sistema turi būti sukurta taikant moduliais-komponentais. | | | | |
| **Pagrindimas:** | Dėmesys teikiamas klasifikavimo algoritmui, jį turi būti galima panaudoti kitose sistemose. | | | | |
| **Šaltinis:** | Studentas | | | | |
| **Atitikimo kriterijus:** | Pagrindinės sistemos teikiamos funkcijos teikiamos naudojant komponentus. | | | | |
| **Užsakovo tenkinimas:** | 5 | **Užsakovo netenkinimas:** | 5 | | |
| **Prioritetas:** | Žemas | **Konfliktai:** | Nėra | | |
| **Istorija:** | Sukurtas: 2024-03-04 | | | | |

### 2.2.6. Reikalavimai saugumui

**36 lentelė** Duomenų validavimo reikalavimas

| | | | | | |
|---|---|---|---|---|---|
| **Reikalavimas #:** | **NR-10** | **Reikalavimo tipas:** | **V15** | **PA:** | **1,3,4,5** |
| **Aprašymas:** | Sistema turi aptikti netinkamų duomenų įvedimą, kur įmanoma tai uždrausti, neleisti įvesti netinkamų atsakymų. | | | | |
| **Pagrindimas:** | Kad duomenys būtų kokybiški pvz. nebūtų atsakymų kaip simbolių seka. Taip pat, apsisaugoti nuo saugumo grėsmių. | | | | |
| **Šaltinis:** | Studentas | | | | |
| **Atitikimo kriterijus:** | Tikrinama informacija gaunama iš laukų kuriuose naudotojas gali laisvai įvesti duomenis. | | | | |
| **Užsakovo tenkinimas:** | 4 | **Užsakovo netenkinimas:** | 5 | | |
| **Prioritetas:** | Žemas | **Konfliktai:** | Nėra | | |
| **Istorija:** | Sukurtas: 2024-03-04 | | | | |

**37 lentelė** Apsaugos nuo kenkėjiško kodo reikalavimas

| | | | | | |
|---|---|---|---|---|---|
| **Reikalavimas #:** | **NR-11** | **Reikalavimo tipas:** | **V15** | **PA:** | **visi** |
| **Aprašymas:** | Sistemoje turi būti realizuoti apsaugos nuo kenkėjiško kodo. | | | | |
| **Pagrindimas:** | Apsaugoti sistemą palaikančia programinę įrangą nuo kenkėjiškų programų. | | | | |
| **Šaltinis:** | Studentas | | | | |
| **Atitikimo kriterijus:** | Realizuoti bent baziniai saugumo patikrinimai į serverį perduodamiems duomenims. | | | | |
| **Užsakovo tenkinimas:** | 3 | **Užsakovo netenkinimas:** | 1 | | |
| **Prioritetas:** | Žemas | **Konfliktai:** | Nėra | | |
| **Istorija:** | Sukurtas: 2024-03-04 | | | | |

**38 lentelė** Slapukų naudojimo reikalavimas

| | | | | | |
|---|---|---|---|---|---|
| **Reikalavimas #:** | **NR-12** | **Reikalavimo tipas:** | **V15** | **PA:** | **visi** |
| **Aprašymas:** | Jei sistemoje bus naudojami slapukai, jų panaudojimas turi atitikti galiojančius teisės aktus ir vartotojų privatumo principus. | | | | |
| **Pagrindimas:** | Sistema negali rinkti ar tvarkyti duomenis pažeidžiant galiojančius įstatymus ir reglamentus. | | | | |
| **Šaltinis:** | Studentas | | | | |
| **Atitikimo kriterijus:** | Sistemoje naudojami slapukai atitinka visus teisės aktus, naudotojas gali pasirinkti kaip bus valdomi jo duomenys. | | | | |
| **Užsakovo tenkinimas:** | 1 | **Užsakovo netenkinimas:** | 4 | | |
| **Prioritetas:** | Žemas | **Konfliktai:** | Nėra | | |
| **Istorija:** | Sukurtas: 2024-03-04 | | | | |

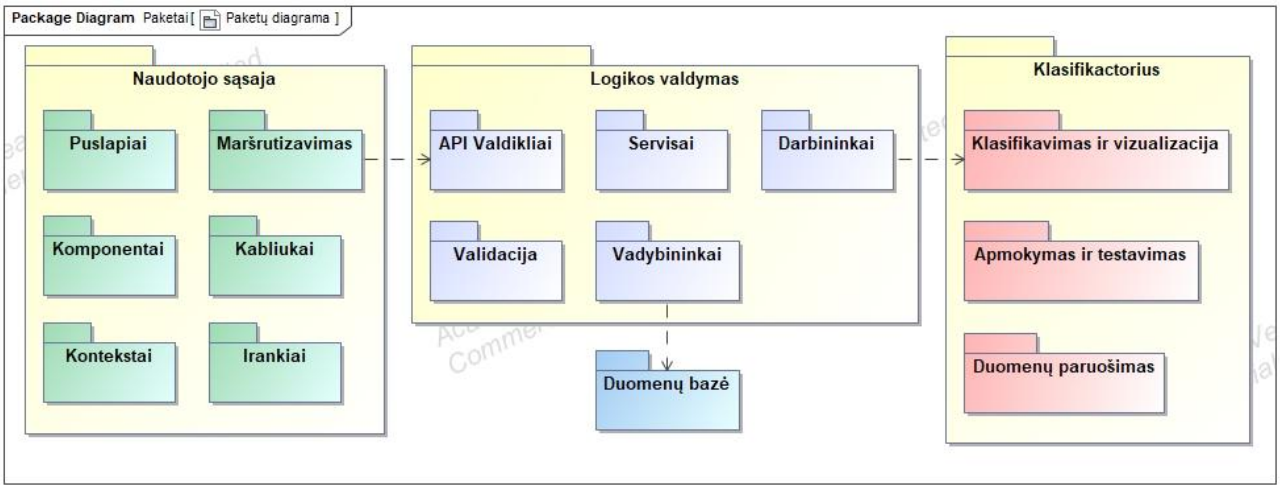
### 2.2.7. Kultūriniai-politiniai reikalavimai

**39 lentelė** Dalykinės kalbos reikalavimas

| | | | | | |
|---|---|---|---|---|---|
| **Reikalavimas #:** | **NR-13** | **Reikalavimo tipas:** | **V16** | **PA:** | **visi** |
| **Aprašymas:** | Informacija sistemoje turi būti pateikta dalykine, gerai suprantama kalba, nenaudojamas<br>žargonas. | | | | |
| **Pagrindimas:** | Sistema yra dalykinio pobūdžio. | | | | |
| **Šaltinis:** | Studentas | | | | |
| **Atitikimo kriterijus:** | Sistemos aprašai, navigacija, informacija pateikta paprasta dalykine lietuvių kalba. | | | | |
| **Užsakovo tenkinimas:** | 4 | **Užsakovo netenkinimas:** | 2 | | |
| **Prioritetas:** | Žemas | **Konfliktai:** | Nėra | | |
| **Istorija:** | Sukurtas: 2024-03-04 | | | | |

### 2.2.8. Sistemos statinis vaizdas

Pateikiamas grafinis sistemos išskaidymas į paketus (žr. 11 pav.).




**11 pav.** Sistemos išskaidymas į paketus

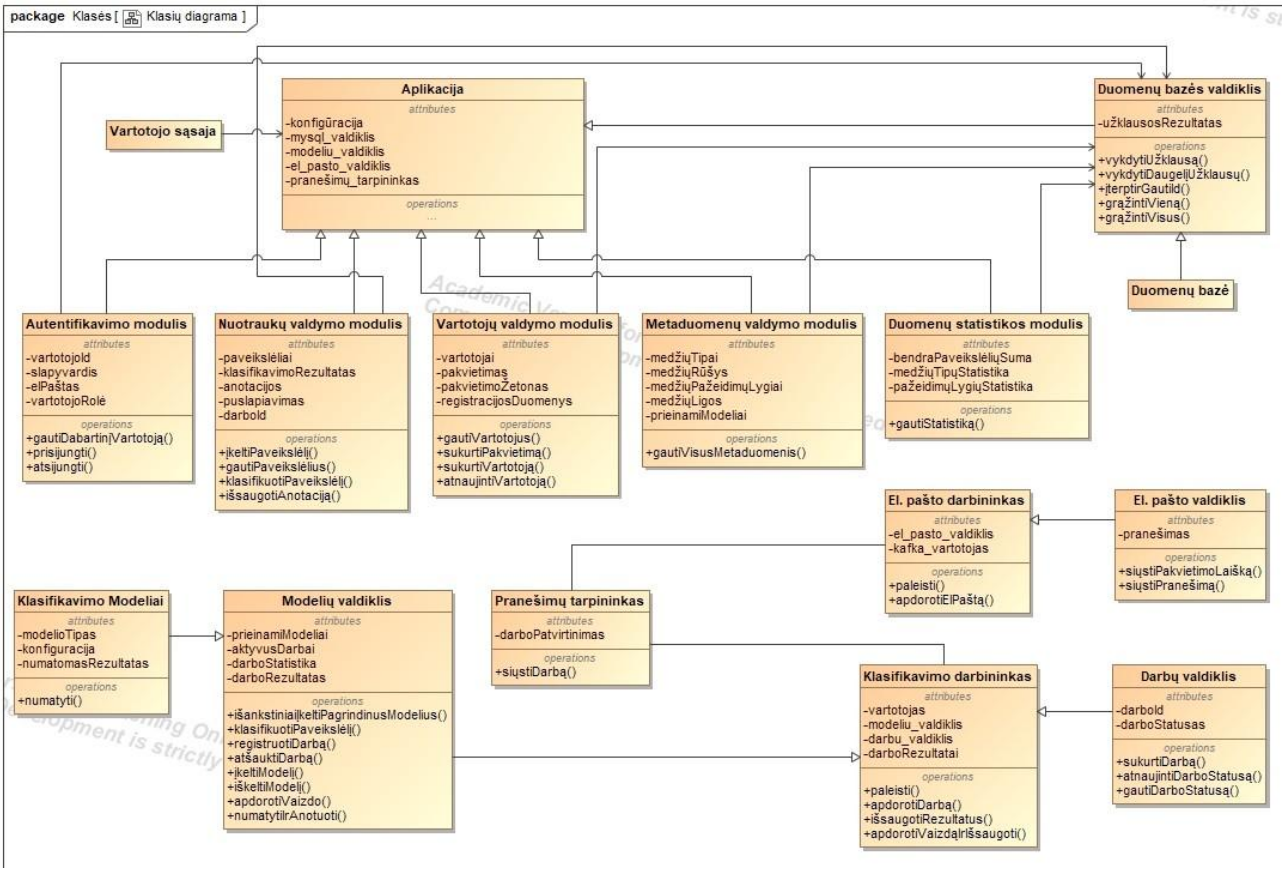
## 2.3. Paketų detalizavimas

### 2.3.1. Paketas „Naudotojo sąsaja“

Šis paketas atsakingas už naudotojo sąsajos atvaizdavimą, paketas grafiškai atvaizduojamas klasių diagrama. Sistemos naudotojo sąsaja kuriama internetinės svetainės forma kuri pasiekiama naudojant populiariąsias naršykles. Naudotojo sąsajai kurti bus naudojama *React* biblioteka. Sistemos funkcionalumas bus pasiekiamas naudojant atskirus tinklalapius atitinkančius panaudojimo atvejus. Visi langai bus perteikti naudojant išdėstymo valdiklį-šabloną, kuris turės nuorodų valdiklį.

### 2.3.2. Paketas „Logikos valdymas“

Paketas skirtas logikos valdymui, pateiktas grafiškai (žr. 12 pav.), naudojamas valdyti naudotojo sąsajos užklausas, taip pat, struktūrizuoti duomenų (skirtų klasifikavimo modeliui apmokyti) kaupimą. Logikos valdymo paketas atitinka vartotojo sąsajos puslapių struktūrą, pridedant duomenų tvarkymo klasę, atsakingą už nuotraukų apdorojimą ir saugojimą serverio pusėje.

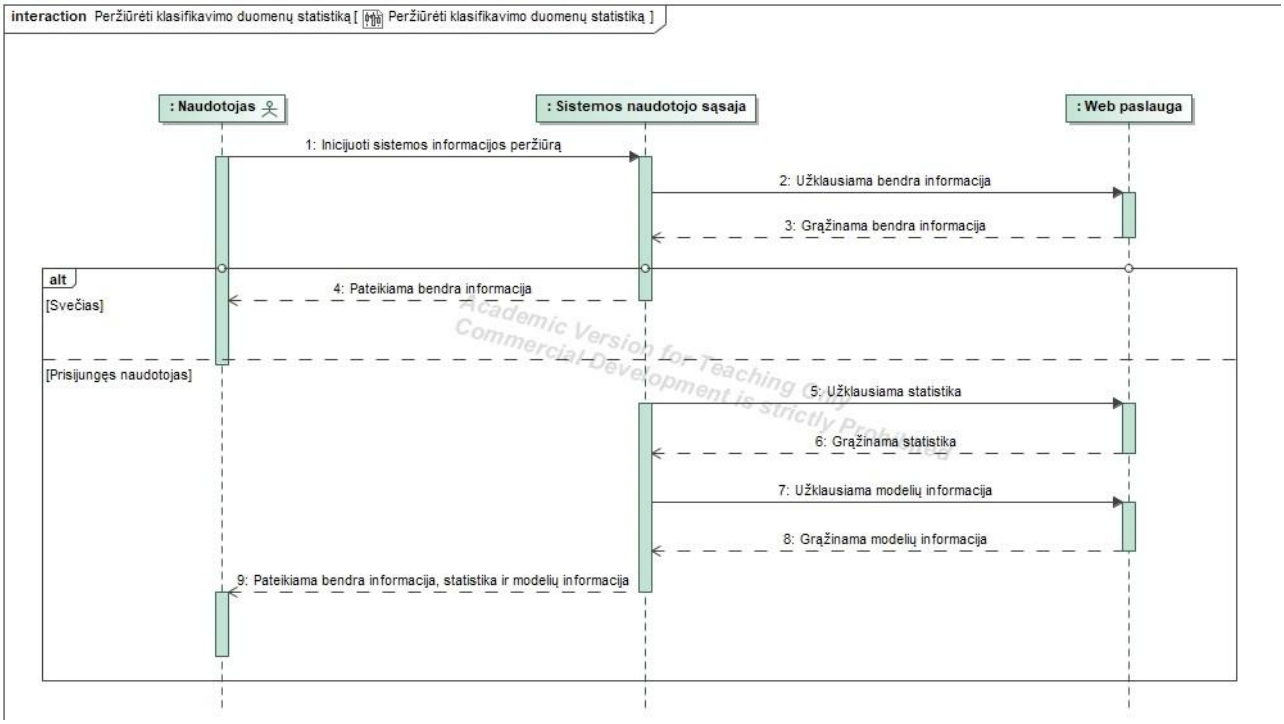



**12 pav.** Paketo „Logikos valdymas“ klasių diagrama

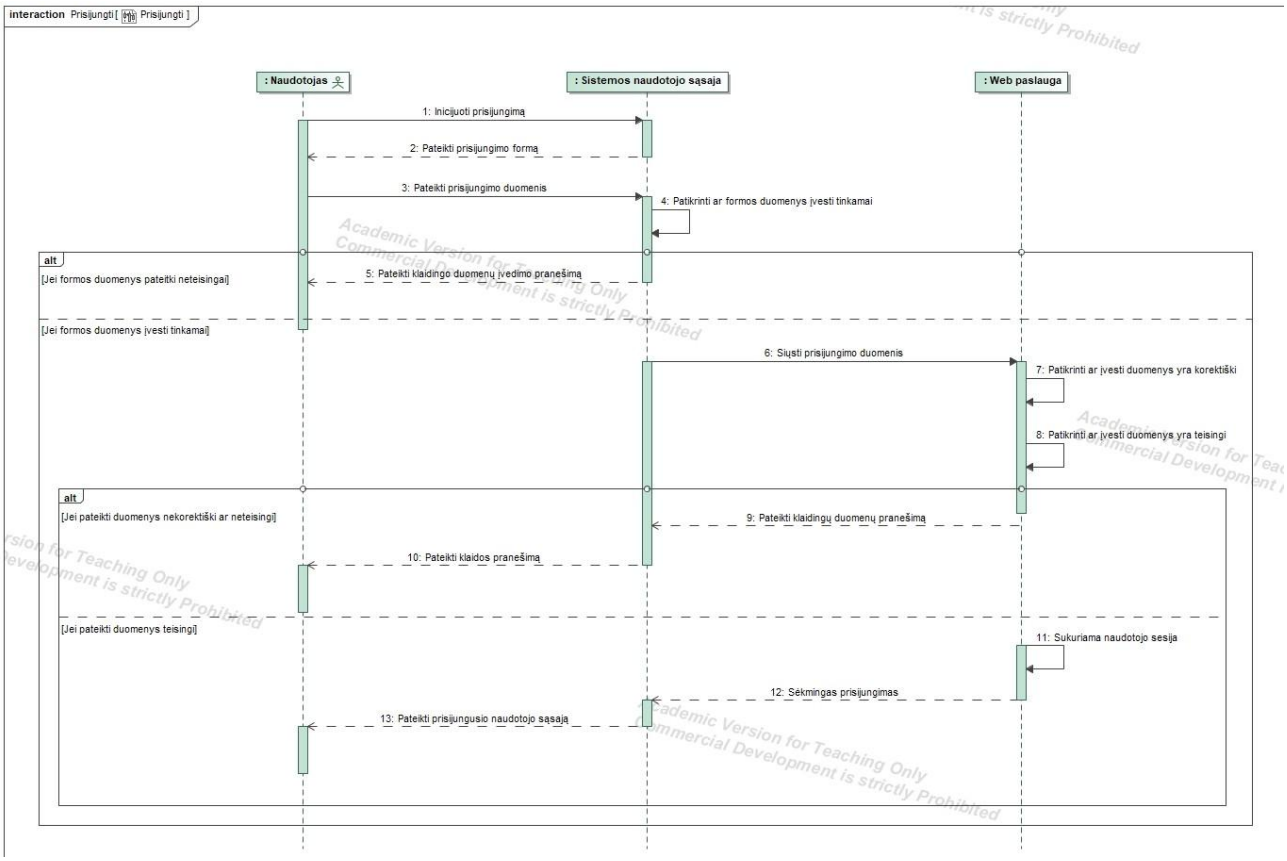
### 2.3.3. Paketas „Klasifikatorius“

Paketas „Klasifikatorius“ yra pagrindinis sistemos paketas, kuriame bus realizuotas CNN algoritmas su pagalbine klase skirta paruošti mokymui skirtas etiketes. Pakete bus realizuotas interfeisas skirtas skirtas apdoroti nuotrauką pateiktą klasifikavimui ir pateikti atsakymą.

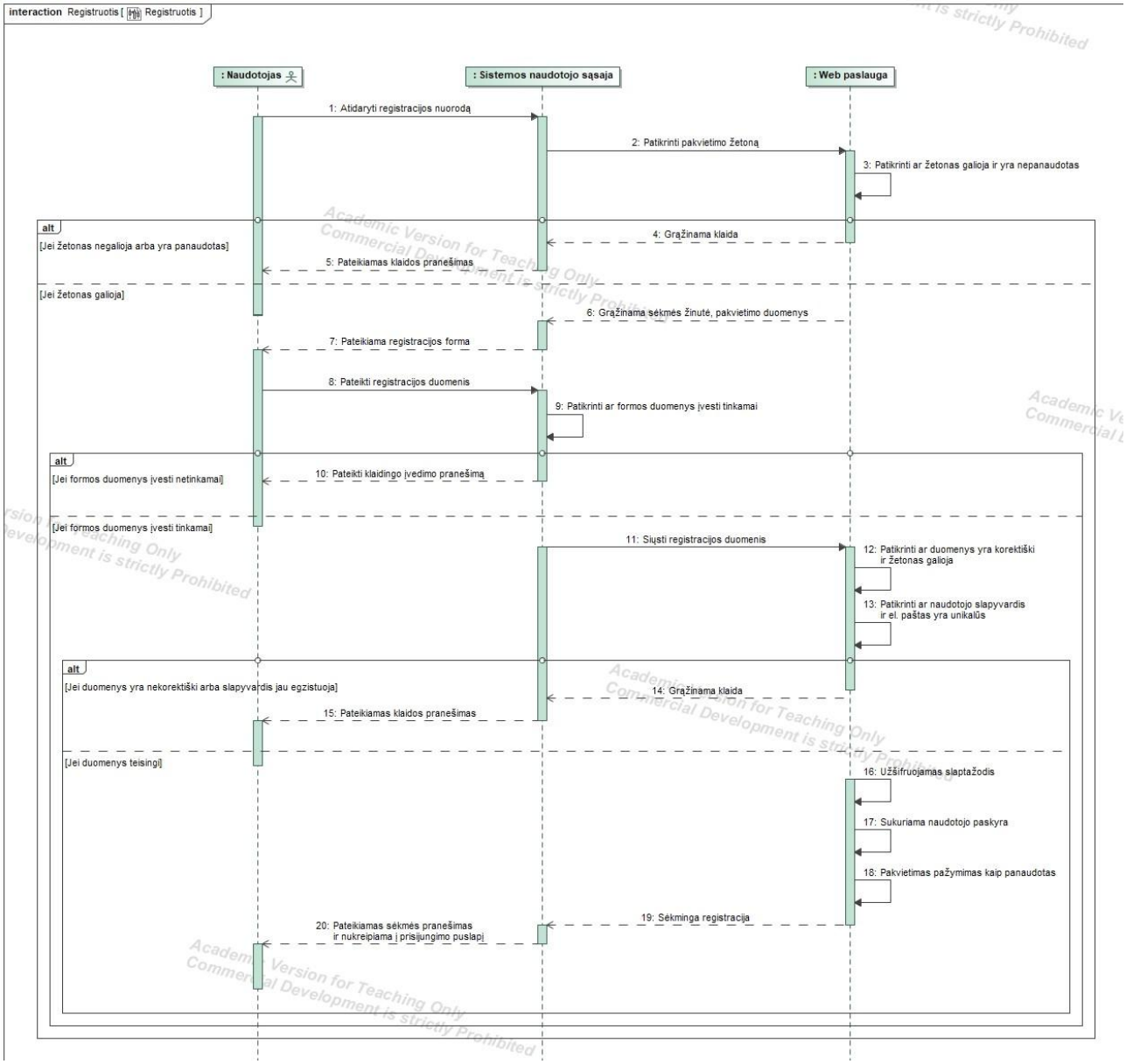
## 2.4. Sistemos dinaminis vaizdas



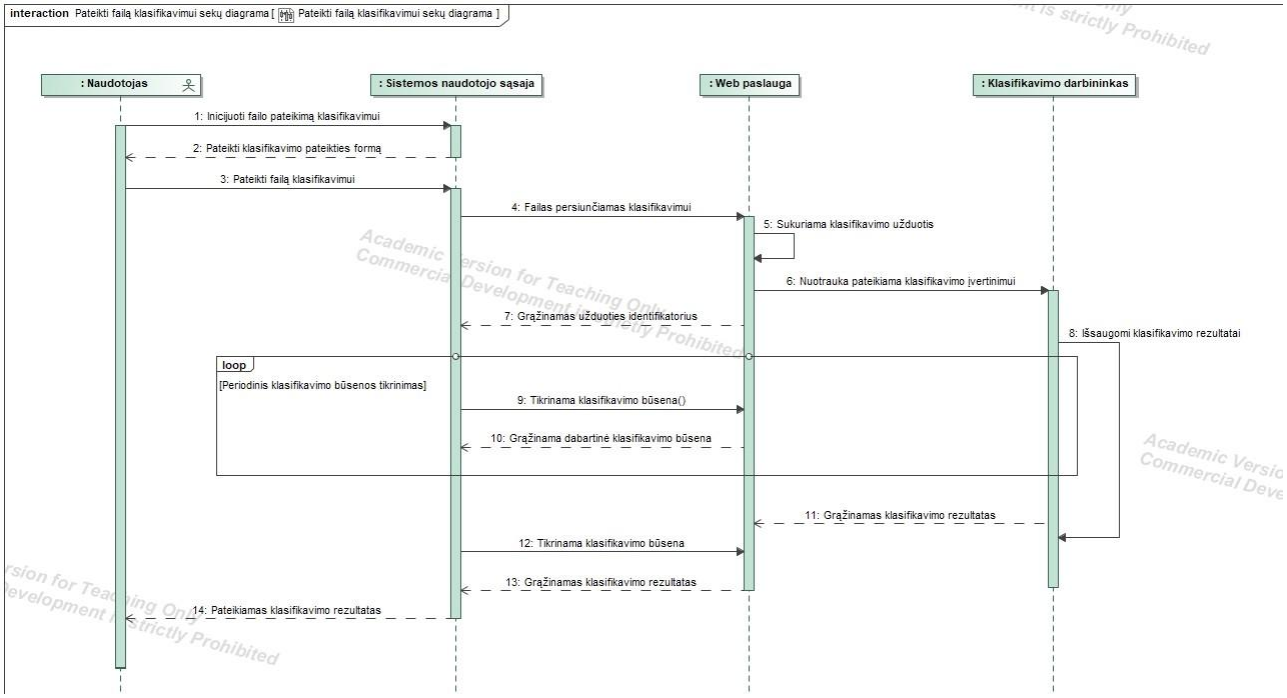

**13 pav.** PA „Peržiūrėti duomenų statistiką“ sekų diagrama



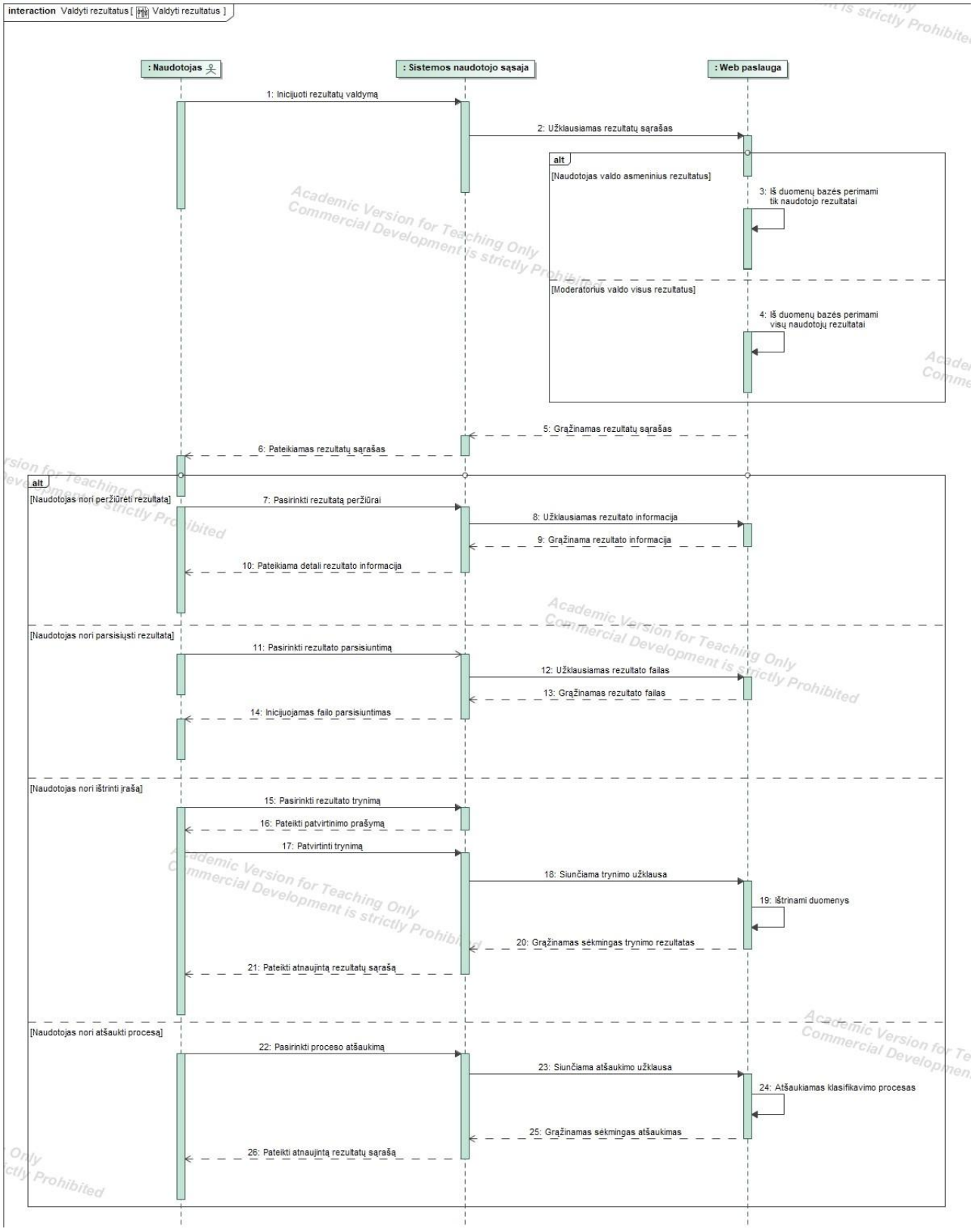

**14 pav.** PA „Prisijungti“ sekų diagrama



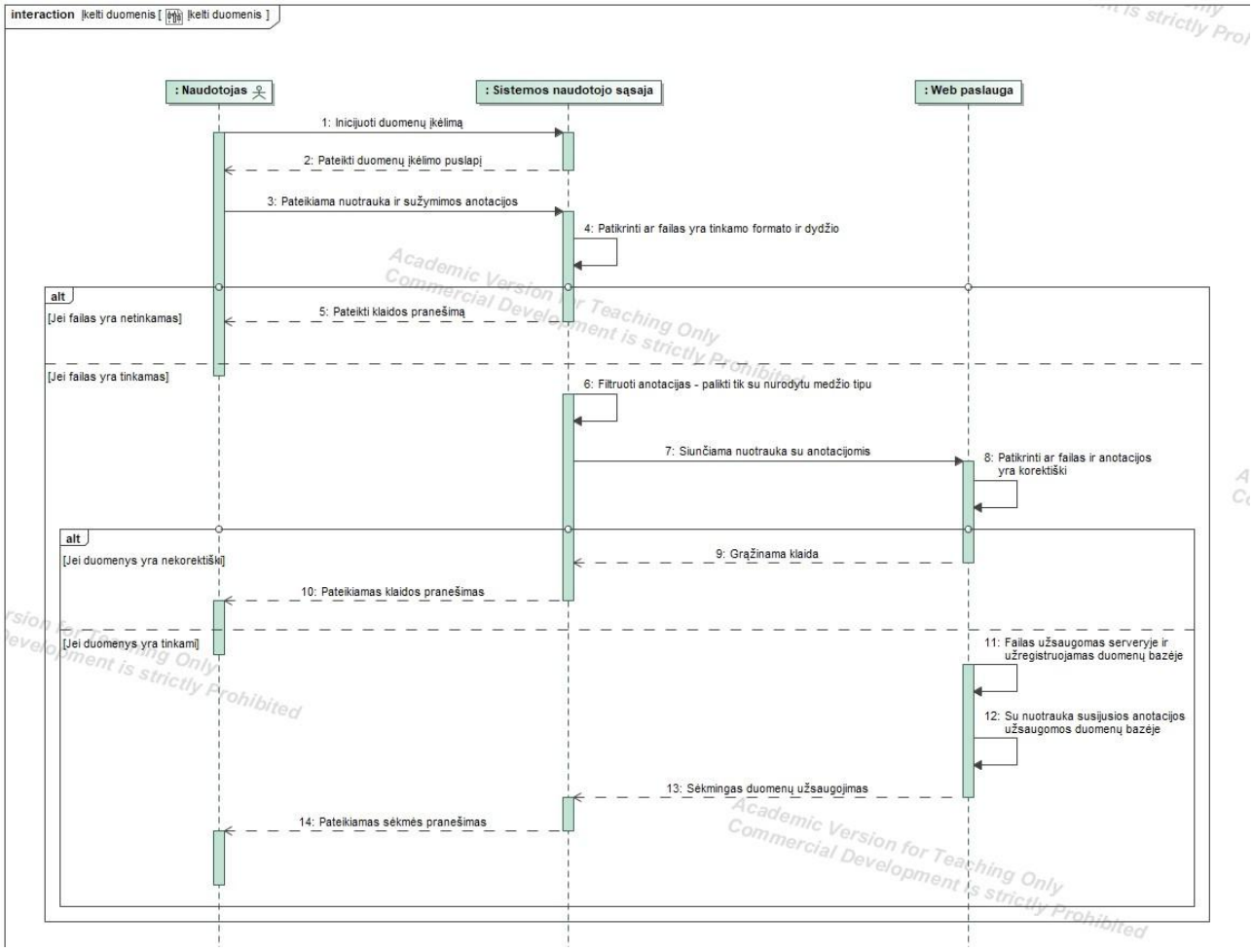

**15 pav.** PA „Registruotis“ sekų diagrama



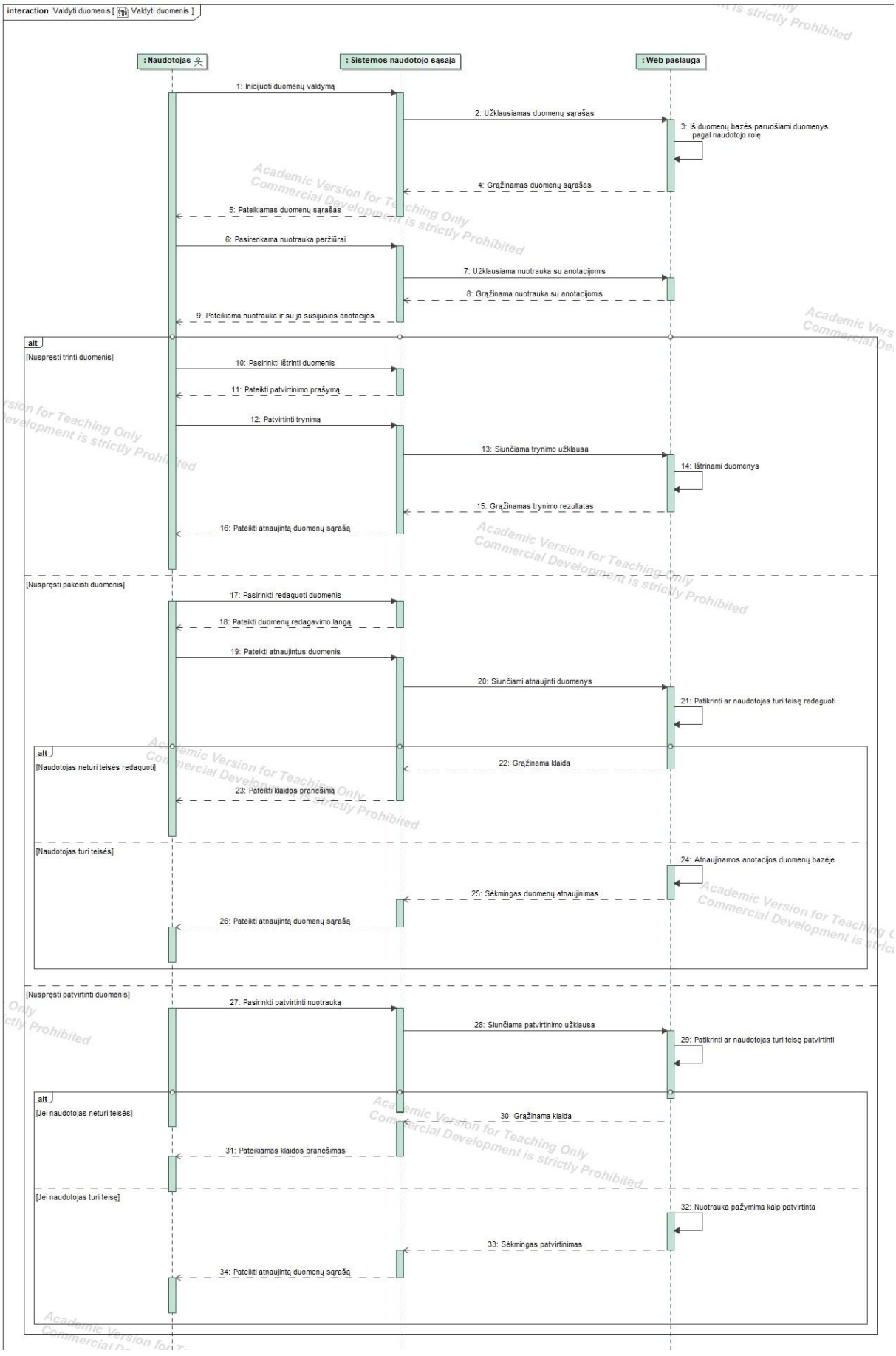

**16 pav.** PA „Pateikti failą klasifikavimui“ sekų diagrama



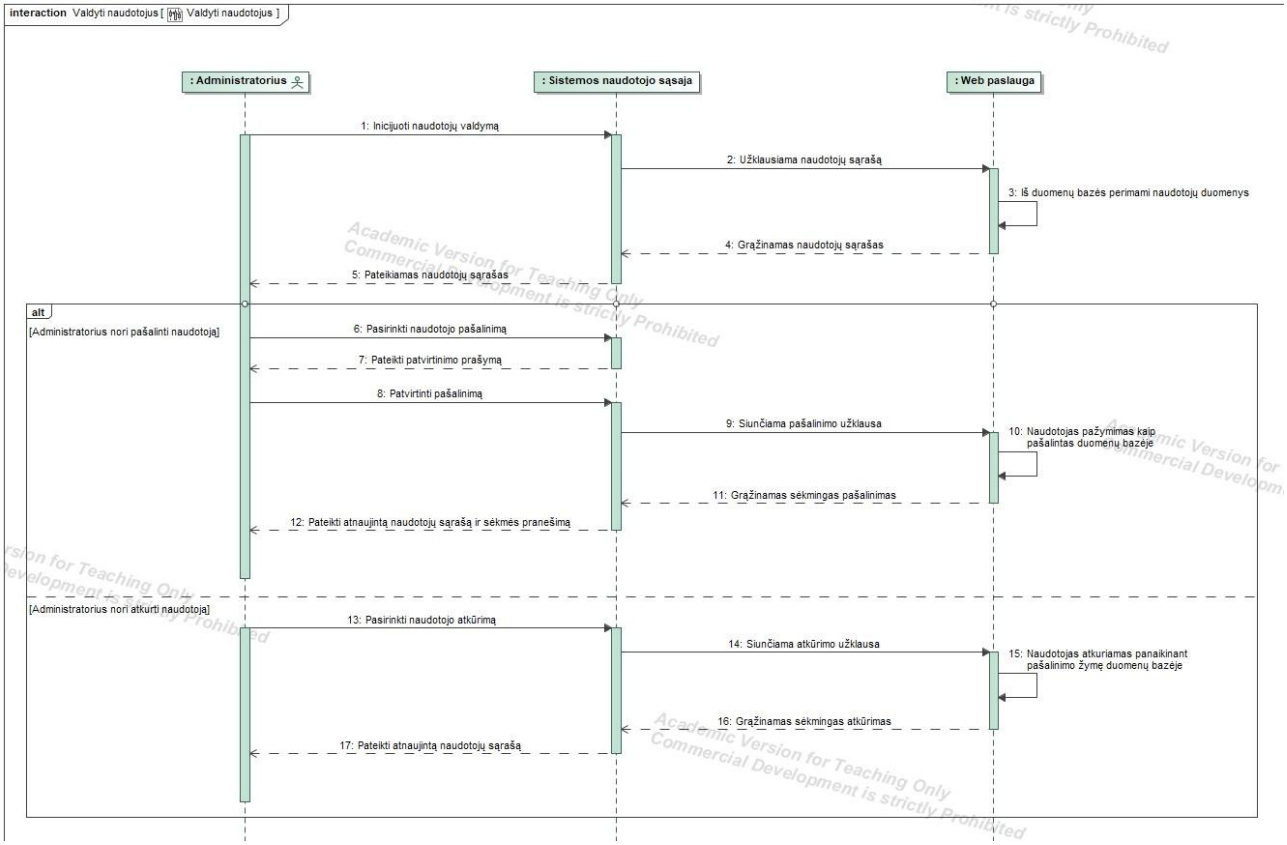

**17 pav.** PA „Valdyti rezultatus“ sekų diagrama



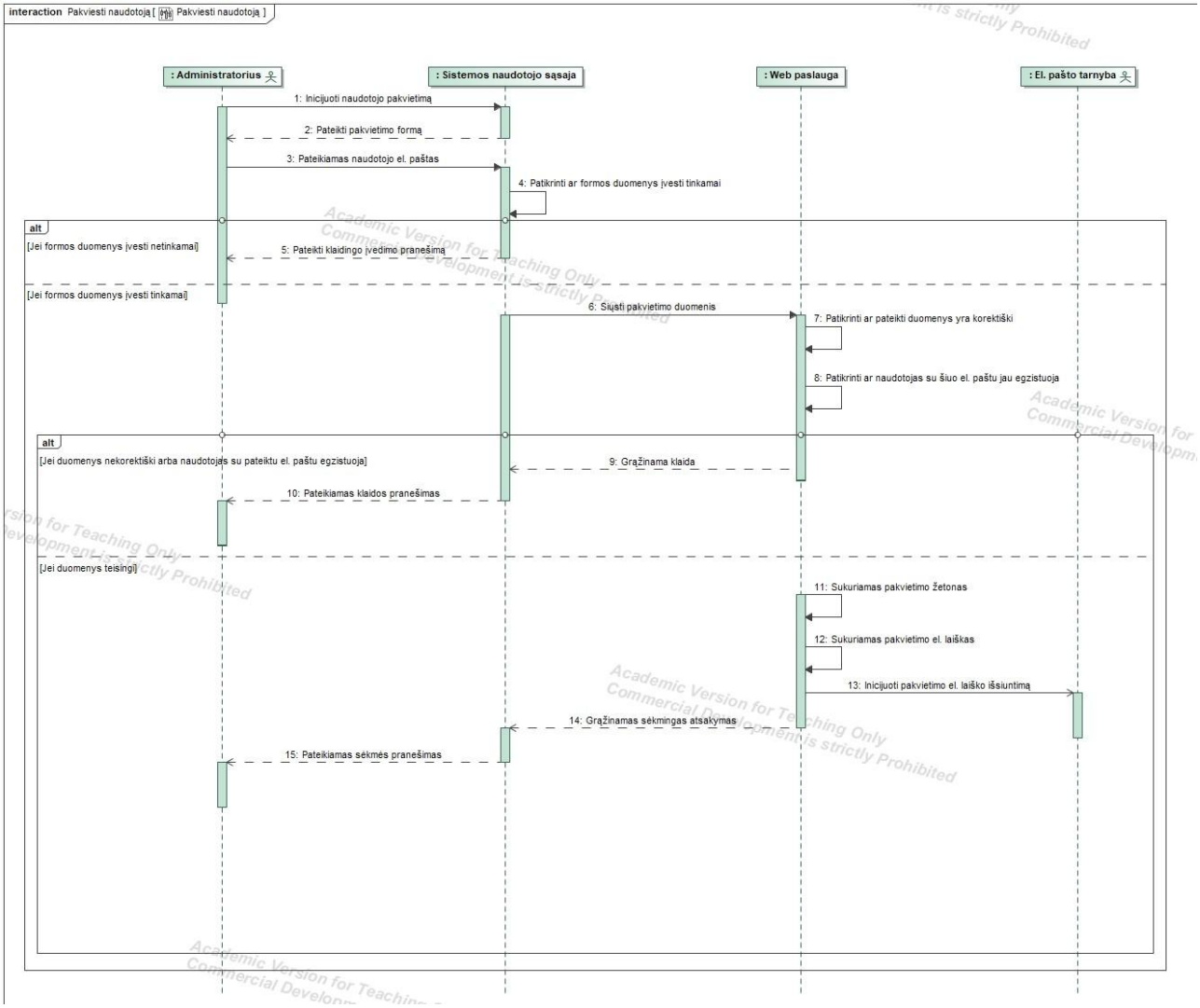

**18 pav.** PA „Įkelti duomenis“ sekų diagrama




**19 pav.** PA „Valdyti duomenis“ sekų diagrama



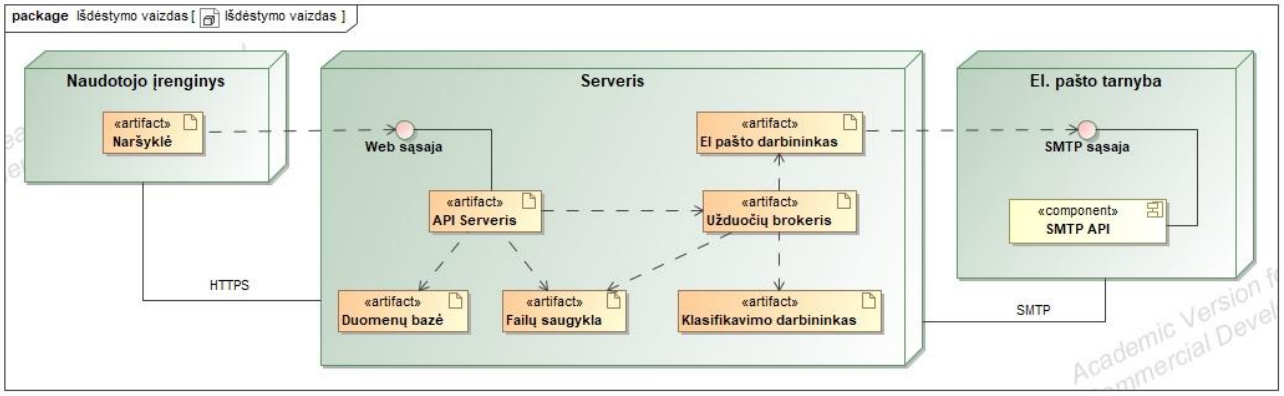

**20 pav.** PA „Valdyti naudotojus“ sekų diagrama




**21 pav.** PA „Pakviesti naudotoją“ sekų diagrama

## 2.5. Išdėstymo vaizdas

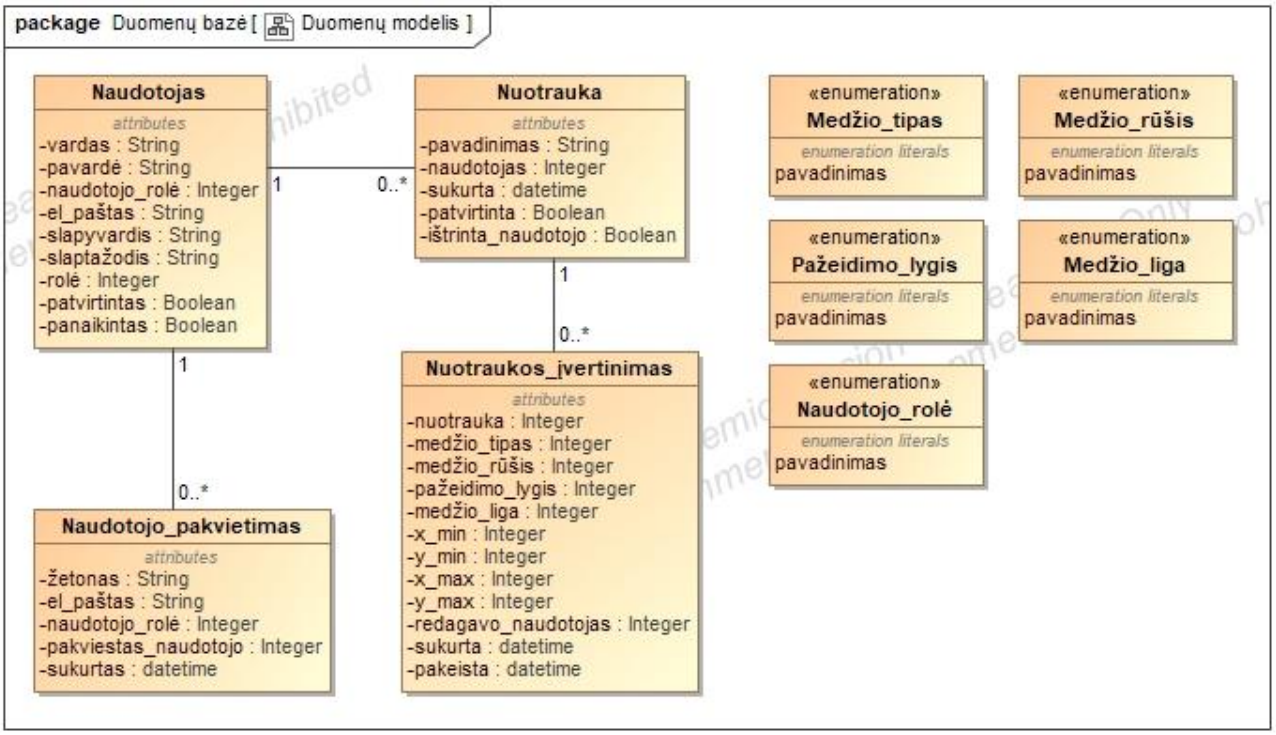
Kuriamos sistemos grafinė sąsaja veiks interneto naršyklėse. Grafinė sąsaja turi veikti naudojant bet kokį asmeninį kompiuterį, siekiamybė – bet kokį įrenginį pvz. išmanųjį telefoną. Naudotojo naudojami įrenginiai turi palaikyti populiariąsias modernias interneto naršykles.




**22 pav.** Sistemos išdėstymo vaizdas

## 2.6. Duomenų vaizdas

Šiame skyriuje pateikiama duomenų modelio diagrama. Kuriamos sistemos duomenys yra saugomi siekiant surinkti duomenų rinkinį skirtą apmokyti neuroninių tinklų klasifikatorių naudojant mašininį mokymą.




**23 pav.** Duomenų modelio diagrama

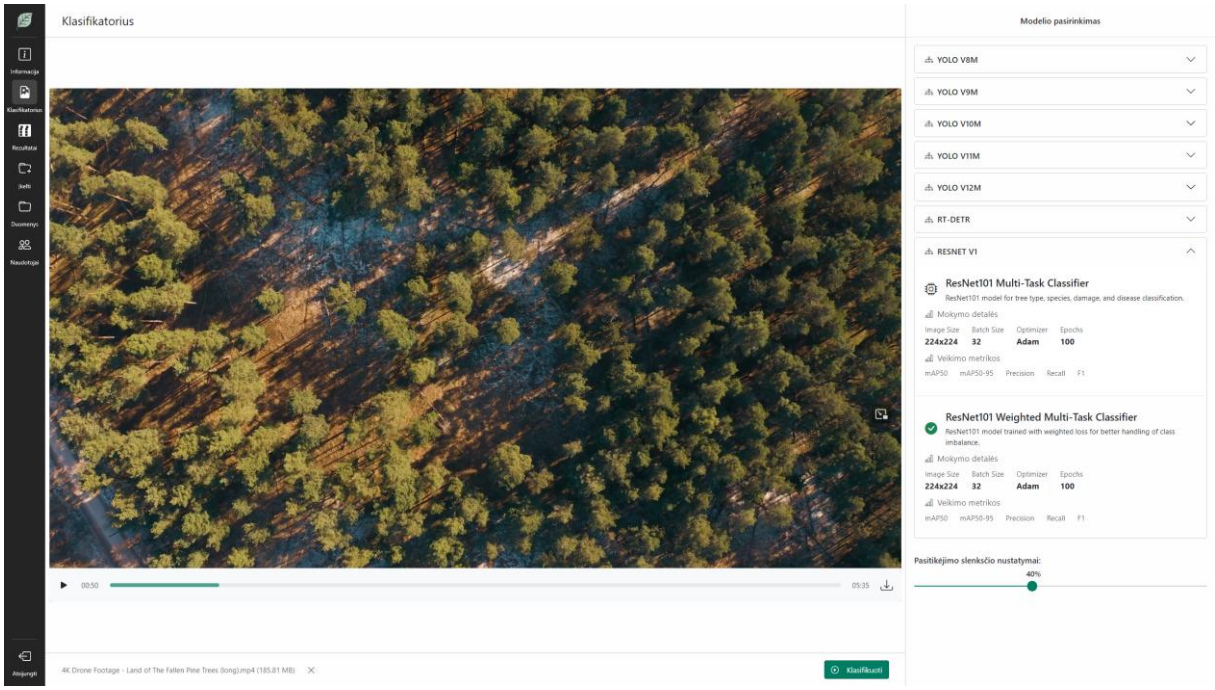
## 2.7. Kokybė

Sistemos architektūra išskaidyta į atskirus komponentus, todėl sistema gali būti išplėsta nesutrikdant pagrindinės sistemos funkcijos – klasifikavimo. Sistema nėra lengvai pernešama, tačiau pagrindinis komponentas – klasifikavimo modelis gali lengvai būti pernešamas, panaudojamas kitose sistemose. Sistemos kūrimas naudojant interneto naršyklės vartotojo sąsają įgalina sukurti vizualiai patrauklią grafinę sąsają, ją suformuoti be apribojimų, tačiau sistemos veikimo patikimumas yra priklausomas nuo naudotojo interneto ryšio. Sistemoje kaupiami apmokymo failai gali būti lengvai pernešami, praplėtus sistemos funkcionalumą, informacija apie nuotraukų informaciją gali būti paverstą į etiketes reikiamas mašininio mokymo algoritmams apmokyti.

## 2.8. Sistemos realizacija

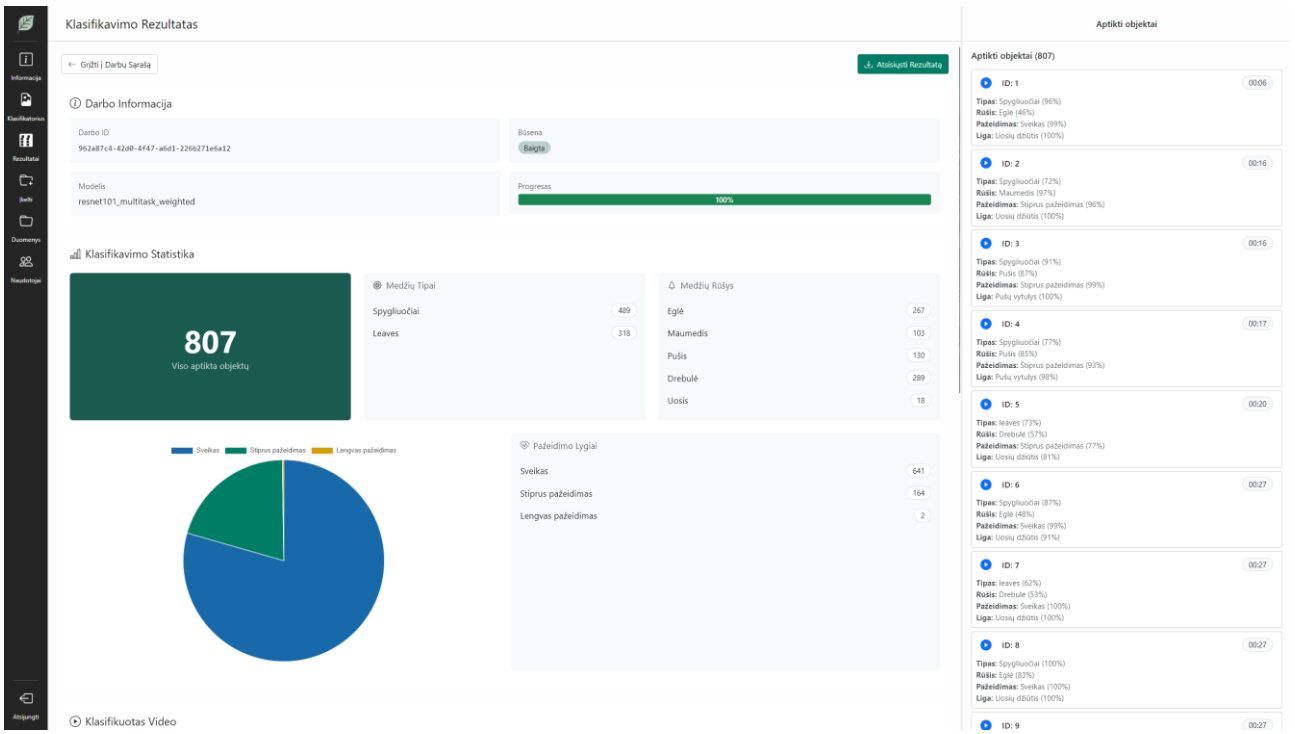
Klasifikatoriaus puslapis (24 pav.) skirtas vaizdinių duomenų (nuotraukų arba vaizdo įrašų) pateikimui automatiniam klasifikavimui, naudojant pasirinktą dirbtinio intelekto modelį. Centrinėje puslapio dalyje pateikiama failų įkėlimo sritis, skirta duomenų pateikimui. Pasirinkus failą, centrinėje puslapio dalyje atvaizduojama pasirinkta nuotrauka arba vaizdo įrašas. Apačioje, dešinėje pusėje esantis mygtukas „Klasifikuoti“ inicijuoja įkelto failo klasifikavimo procesą. Kairėje pusėje esantis mygtukas su piktograma „X“ leidžia pašalinti įkeltą failą prieš pradedant analizę. Dešinėje puslapio pusėje pateikiama šoninė juosta,

skirta modelio pasirinkimui ir objektų aptikimo pasitikėjimo slenksčio nustatymui. Inicijavus klasifikavimo procesą, sistema automatiškai nukreipia į rezultatų peržiūros puslapį.

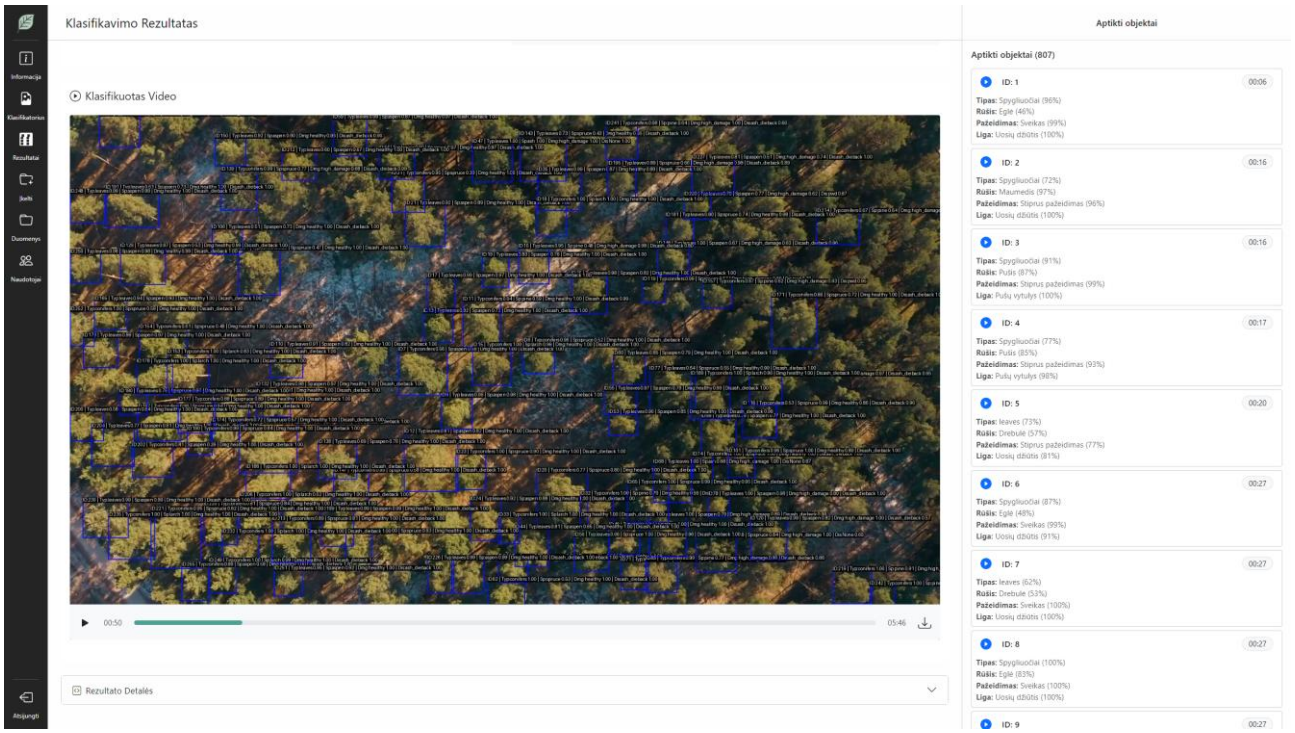



**24 pav.** Klasifikavimo puslapis – po failo ir klasifikavimo modelio pasirinkimo

Klasifikavimo rezultatų puslapis (žr. 25–26 pav.) skirtas vaizdinių duomenų klasifikavimo rezultatų peržiūrai. Puslapyje pateikiama detali klasifikavimo suvestinė, kurioje rodoma statistika apie aptiktus objektus, vizualiai atvaizduojami rezultatai ir pateikiamas objektų sąrašas su skaitiniais klasifikavimo įvertinimais. Viršutinėje puslapio dalyje esanti informacinė juosta pateikia pagrindinius valdymo elementus ir darbo informaciją: mygtuką grįžti į atliktų darbų sąrašą, mygtuką rezultatų atsisiuntimui, darbo identifikacinį numerį, naudotą klasifikavimo modelį, proceso eigos statusą ir būseną. Centrinėje puslapio dalyje rodoma klasifikavimo statistika (žr. 25 pav.) ir vizualizuotas rezultatas (žr. 26 pav.) – nuotraukose objektai sužymėti interaktyviais aptikimo rėmeliais, vaizdo įrašams naudojamas grotuvas, kuriame galima peržiūrėti sužymėtą turinį. Dešinėje puslapio pusėje pateikiamas aptiktų objektų sąrašas su skaitiniais klasifikavimo duomenimis; pasirinkus objektą sąraše, jis paryškinamas vizualizacijoje arba grotuvas persukamas į vaizdo įrašo momentą, kai objektas buvo aptiktas.




**25 pav.** Klasifikavimo rezultatų peržiūros puslapis – rezultatų suvestinė




**26 pav.** Klasifikavimo rezultatų peržiūros puslapis – vizualiai sužymėtų objektų peržiūra

# 3. Tyrimas

Šio tyrimo tikslas – ištirti ir įvertinti kompiuterinės regos metodų, grįstų giluminiu mokymusi, taikymo galimybes automatiniam miško medžių lajų pažeidimų identifikavimui naudojant nepilotuojamų orlaivių užfiksuotus vaizdus. Eksperimento metu išbandomos ir lyginamos dvi objektų aptikimo ir klasifikavimo strategijos: vienos stadijos ir dviejų stadijų. Vienos stadijos strategija tyrime naudoja vieną modelį atlikti objektų aptikimą ir klasifikavimą kartu. Dviejų stadijų strategija atskiria objektų aptikimą ir klasifikavimą tam naudojant atskirus modelius. Dviejų stadijų strategijoje, pirmajame etape objektų aptikimo modelis aptinka medžius kaip vieną bendrą klasę, antrajame etape kiekviena aptikta medžio sritis iškerpama ir perduodama objektų klasifikatoriui, kuris ją priskiria vienai iš klasių.

Abi nagrinėjamos architektūros turi savo stipriųjų ir silpnųjų pusių, susijusių su medžių pažeidimų klasifikavimu nuotraukose. Vienos stadijos požiūriu pagrįsti modeliai veikia greičiau, nes reikalingas tik vienas praėjimas per vaizdą, tačiau jie apsiriboja viena klasifikavimo užduotimi. Dviejų stadijų strategija pagrįstos architektūros gali suteikti daugiau informacijos, pvz. klasifikuoti pagal medžio tipą, rūšį, pažeidimo lygį, ligą, tačiau jiems reikia dviejų atskirų modelių ir daugiau skaičiavimo išteklių.

Tyrimo metu atliekamais eksperimentais siekiama įvertinti abiejų strategijų klasifikavimo tikslumą ir laiko sąnaudas nustatant miško medžių pažeidimo lygį. Taip pat, patikrinti, ar dviejų stadijų strategija gali pranokti vienos stadijos sprendimo rezultatus.

## 3.1. Naudojamų modelių aprašymas

### 3.1.1. Objektų aptikimo modeliai

Tyrimo metu, objektų aptikimui išbandomi keli modeliai, kurie naudoja skirtingas architektūras. Šios architektūros atstovauja klasikinių konvoliucinių neuroninių tinklų, vienos stadijos aptikimo ir transformerių architektūra veikiančius metodus, leidžiančius išsamiai įvertinti skirtingų detektorių tikslumą ir efektyvumą. Objektų aptikimui išbandyti šie modeliai:

- Faster R-CNN [27];
- YOLO [28];
- RT-DETR [30];
- RF-DETR [31].

Faster R-CNN yra klasikinis, konvoliuciniais neuroniniais tinklais pagrįstas dviejų stadijų detektorius, kurio pirmoje stadijoje regionų pasiūlymų tinklas (angl. *Region Proposal Network*) generuoja kandidatinius regionus, o antroje stadijoje kiekvienas regionas yra klasifikuojamas ir objekto ribas nustatantis stačiakampis patikslinamas [27]. Faster R-CNN yra vienas pirmųjų konvoliucinius neuroninius tinklus naudojantis modelis, tačiau vis dar naudojamas praktikoje.

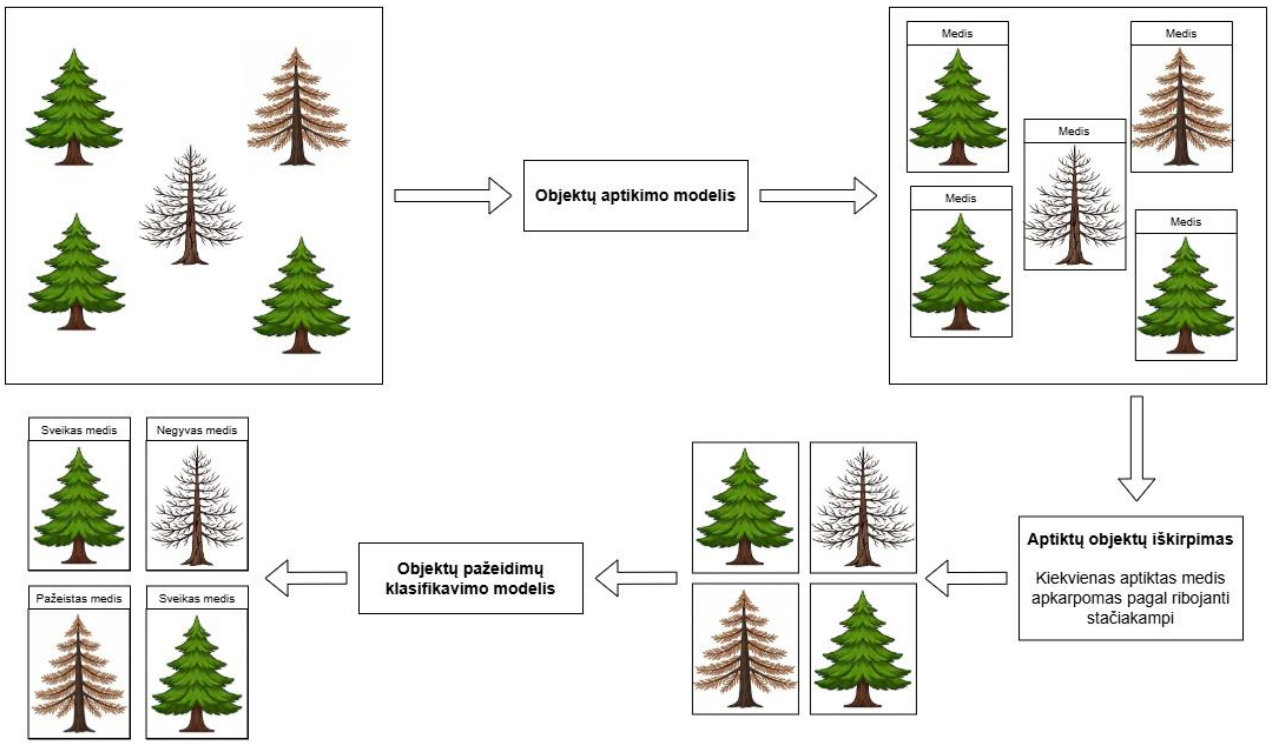
YOLO yra viena populiariausių vienos stadijos gilaus mokymosi architektūrų, plačiai taikoma realaus laiko objektų aptikimo užduotyse. Ją sudaro trys pagrindiniai komponentai: „*CSPDarknet*“ pagrindas, išgaunantis požymius iš įvesties vaizdo; „*FPN+PAN*" kaklas, apjungiantis skirtingų rezoliucijų požymių žemėlapius ir aptikimo galva, prognozuojanti objekto ribas nustatančių stačiakampių koordinates bei klasių tikimybes [28]. Toks modulinis

architektūros padalijimas leidžia efektyviai aptikti įvairaus dydžio objektus viename vaizdo apdorojimo žingsnyje, todėl YOLO modeliai pasižymi aukštu tikslumu ir greitu veikimu.

RT-DETR (angl. *Real-Time Detection Transformer*) yra transformerių pagrindu sukurtas objektų aptikimo modelis, kuris nuo YOLO skiriasi esminiu veikimo principu – vietoje iš anksto apibrėžtų inkarinių dėžių (angl. *anchor boxes*) RT-DETR taiko dėmesio mechanizmus, leidžiančius modeliui vienu metu vertinti visus vaizde esančius objektus ir jų tarpusavio ryšius [30]. Toks požiūris teoriškai turėtų pagerinti aptikimo tikslumą, tačiau transformerių modeliai apmokymui reikalauja daugiau duomenų ir skaičiavimo išteklių nei CNN pagrindu sukurti modeliai. RF-DETR (angl. *Receptive Field DETR*) yra naujesnės kartos transformerių detektorius, optimizuotas mažiems objektams aptikti [31]. Skirtingai nuo RT-DETR, jis taiko deformuojamus dėmesio mechanizmus su adaptyviais imtuvų laukais, kurie leidžia modeliui dinamiškai pritaikyti dėmesio sritį pagal objekto dydį ir formą. Ši savybė ypač aktuali aerovaizdams, kuriuose medžių objektai vaizde gali būti labai skirtingų dydžių.

### 3.1.2. Dviejų stadijų architektūra

Dviejų stadijų klasifikavimo architektūra tyrime sujungia objektų aptikimo ir klasifikavimo modelius į vieną sistemą (žr. 27 pav.). Pirmajame etape pilnas aeronuotraukos vaizdas perduodamas objektų aptikimo modeliui, apmokytam aptikti visus medžius kaip vieną bendrą klasę gautą sujungus visas pažeidimų klases. Antrajame etape kiekvienam aptiktam medžiui vaizdas apkarpomas pagal ribojantį stačiakampį ir normalizuojamas. Tuomet objektų klasifikavimo modelis kiekvieną apkarpytą medžio vaizdą priskiria vienai iš 4 pažeidimų klasių.




**27 pav.** Dviejų stadijų klasifikavimo strategija

Dviejų stadijų klasifikavimo procesas:

1. pateikties vaizdas yra perduodamas objektų aptikimo modeliui, apmokytam aptikti visus medžius nepriklausomai nuo pažeidimo lygio;
2. kiekvienas aptiktas objektas yra iškerpamas iš pateikties vaizdo naudojant objektų ribų stačiakampius;
3. iškirptas objektas-vaizdas yra perduodamas pateikiamas klasifikavimo modeliui kuris nustato kuriai klasei iškirptas objektas priklauso.

Toks atskirų specializuotų modelių derinimas suteikia galimybę kiekvieną stadiją optimizuoti atskirai pagal jos užduoties specifiką. Tačiau ši architektūra turi ir esminį trūkumą: pirmojoje stadijoje padarytos klaidos, pavyzdžiui, neaptikti arba klaidingai aptikti medžiai, tiesiogiai perkeliamos į antrąją stadiją ir joje nebegali būti ištaisytos.

### 3.1.3. Objektų klasifikavimo modeliai

Dviejų stadijų požiūrio antroje stadijoje naudojami konvoliucinių neuroninių tinklų architektūra paremti klasifikavimo modeliai. Eksperimente šie modeliai objektų aptikimo modelių nustatytoms ir iškirptoms vaizdų sritims priskiria atitinkamą klasę. Tyrimo metu yra išbandomos kelios skirtingos architektūros, kurios pasižymi skirtingais projektavimo principais:

- ResNet101 [22];
- EfficientNetV2 [23];
- ConvNeXtV2 [24].

ResNet101 yra gili CNN architektūra, kurios pagrindinė savybė – liekamieji ryšiai, leidžiantys efektyviai perduoti gradientus (angl. *gradients*) per tinklą ir sumažinti gradiento nykimo (angl. *vanishing gradient*) problemą. Modelis susideda iš požymių išgavimo pagrindo apmokyto ImageNet [32] duomenų rinkiniu ir klasifikavimo sluoksnio, kuris išgautus požymius atvaizduoja į tikslinių klasių tikimybes. Toks sprendimas leidžia efektyviai išnaudoti iš anksto išmoktus vizualinius požymius naujoje užduotyje [22].

EfficientNetV2 yra CNN architektūra, kurios pagrindinė savybė – subalansuotas tinklo gylio, pločio ir įvesties rezoliucijos mastelio keitimas (angl. *compound scaling*) bei optimizuoti konvoliuciniai blokai, leidžiantys pasiekti aukštą tikslumą su mažesniu parametrų skaičiumi ir trumpesniu mokymo laiku. Kaip ir ResNet101, modelis susideda iš požymių išgavimo pagrindo apmokyto ImageNet duomenų rinkiniu ir klasifikavimo sluoksnio. Toks sprendimas leidžia efektyviai išnaudoti anksčiau išmoktus vizualinius požymius ir tinka ribotų skaičiavimo išteklių scenarijams [23].

ConvNeXtV2 yra modernizuota CNN architektūra, kurios pagrindinė savybė – transformerių principais perprojektuoti vidiniai tinklo komponentai, leidžiantys pasiekti aukštą tikslumą su panašiomis skaičiavimo sąnaudomis kaip ir tradicinės CNN architektūros. Kaip ir ankstesnių modelių atveju, ConvNeXtV2 taiko perdavimo mokymosi principą – ImageNet duomenų rinkiniu apmokytas požymių išgavimo pagrindas yra papildomas tikslinei užduočiai pritaikytu klasifikavimo sluoksniu [24].

## 3.2. Duomenų paruošimas ir analizė

### 3.2.1. Duomenų rinkinio apžvalga

Pirmajame tyrimo etape yra naudojamas „Forest Damages – Larch Casebearer" duomenų rinkinys [33], sudarytas iš drono nuotraukų, kuriuose anotuoti maumedžių pažeidimų lygiai. Duomenų rinkinys anotuotas YOLO formatu su 4 klasėmis, atitinkančiomis maumedžių sveikatos lygius ir kitų rūšių medžius pateikiamas 40 lentelėje. Pažeidimų lygiai suskirstyti į: sveikas, lengvai pažeistas, sunkiai pažeistas ir kitas (visų kitų rūšių medžiai). Šis rinkinys pasirinktas nes duomenys yra sužymėti profesionalų, dalykinės srities ekspertų, todėl žymėjimo kokybę galima laikyti aukšta. Tačiau, pagrindinis duomenų rinkinio trūkumas – nedidelis anotuotų vaizdų skaičius (840), kuris gali riboti modelio apibendrinimo gebėjimus.

**40 lentelė** Pradinis duomenų rinkinio klasių pasiskirstymas

| Klasė | Atitinka | Objektų sk. | Dalis |
|---|---|---|---|
| H | Sveikas | 2 638 | 4,67 % |
| LD | Lengvai pažeistas | 33 299 | 58,97 % |
| HD | Sunkiai pažeistas | 9 086 | 16,09 % |
| OTHER | Kiti | 11 400 | 20,19 % |
| Iš viso | | 56 423 | 100 % |

Pasirinkto duomenų rinkinio, prieš apdorojimą, objektų klasių pasiskirstymas atvaizduojamas grafiškai stulpeline diagrama (žr. 28 pav.).

**28 pav.** Duomenų rinkinio klasių pasiskirstymas

Iš lentelės ir stulpelinės diagramos matyti, kad objektų pasiskirstymas duomenų rinkinyje nėra tolygus – egzistuoja ženklus klasių disbalansas. Duomenų rinkinyje dominuoja lengvo pažeidimo klasės objektai, kurių yra daugiau nei kitų klasių kartu sudėjus. Duomenų pažymėtų kaip sveikas medis yra mažiausiai ir reikšmingai mažiau nei kitų tipų objektų. Nors

klasių disbalansas kelia iššūkių modelio mokymui, šis rinkinys buvo pasirinktas, nes yra kokybiškiausias prieinamas šaltinis.

### 3.2.2. Duomenų augmentacija ir pirminis apdorojimas

Prieš modelių apmokymą duomenims buvo taikomos augmentacijos technikos, skirtos padidinti duomenų įvairovę ir sumažinti persimokymą. Duomenų rinkinys buvo padalintas į 70 % mokymui, 20 % validavimui ir 10 % testavimui dalis – tai yra plačiai taikomas skaidymo santykis. Validavimo ir testavimo duomenys nebuvo išplėsti (angl. *augmented*) siekiant „nesuteršti" validavimo ir testavimo vertinimo proceso ir užtikrinti, kad modelio rezultatai būtų tikrinami su realiais, nemodifikuotais duomenimis. Mokymo duomenų daliai buvo taikyta augmentacija – keturi pasukimai ir horizontalus atspindys – siekiant padidinti mokymo aibės dydį ir įvairovę. Galutinis vaizdų ir objektų skaičius kiekvienoje aibėje pateikiamas 41 lentelėje.

**41 lentelė** Duomenų rinkinio struktūra

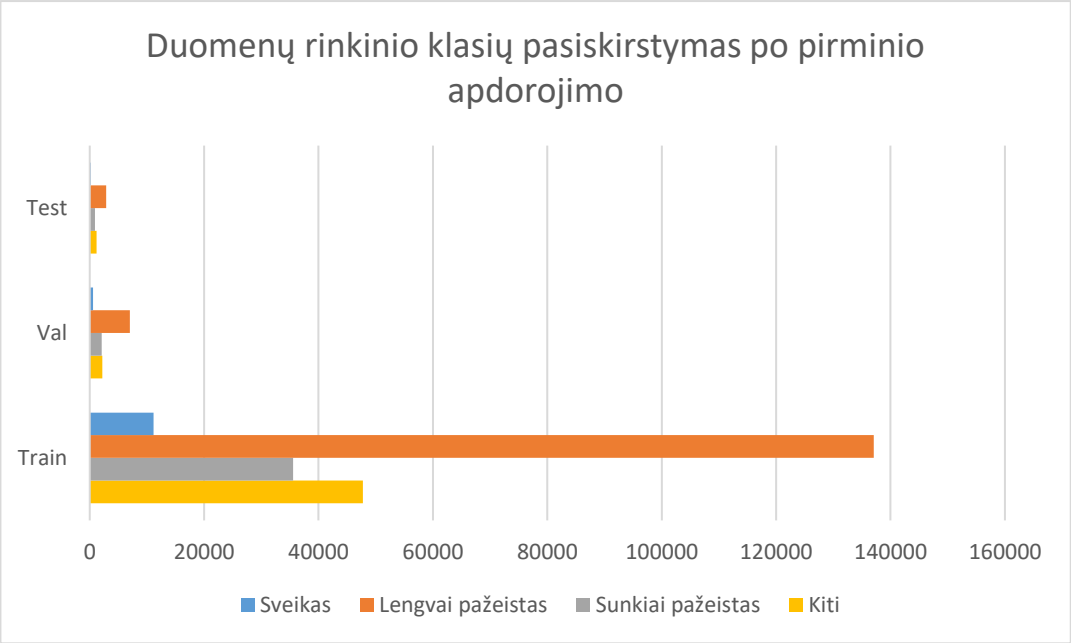
| **Rinkinys** | **Vaizdų sk.** | **Objektų sk.** |
|---|---|---|
| Mokymo | 3 510 | 231 529 |
| Validavimo | 167 | 11 858 |
| Testavimo | 85 | 5 132 |
| Iš viso | 3 762 | 248 519 |

Atlikus augmentaciją, mokymo aibė išaugo nuo 588 iki 3 510 vaizdų – beveik 6 kartus. Dėl šios priežasties mokymo aibėje duomenų yra keliskart daugiau nei validavimo (167 vaizdai) ir testavimo (85 vaizdai) aibėse, kurios nebuvo augmentuojamos ir išliko nepakitusios (26 pav.). Objektų pasiskirstymas pagal klases po duomenų apdorojimo pateikti 42 lentelėje.

**42 lentelė** Duomenų rinkinio objektų pasiskirstymas atlikus duomenų apdorojimą

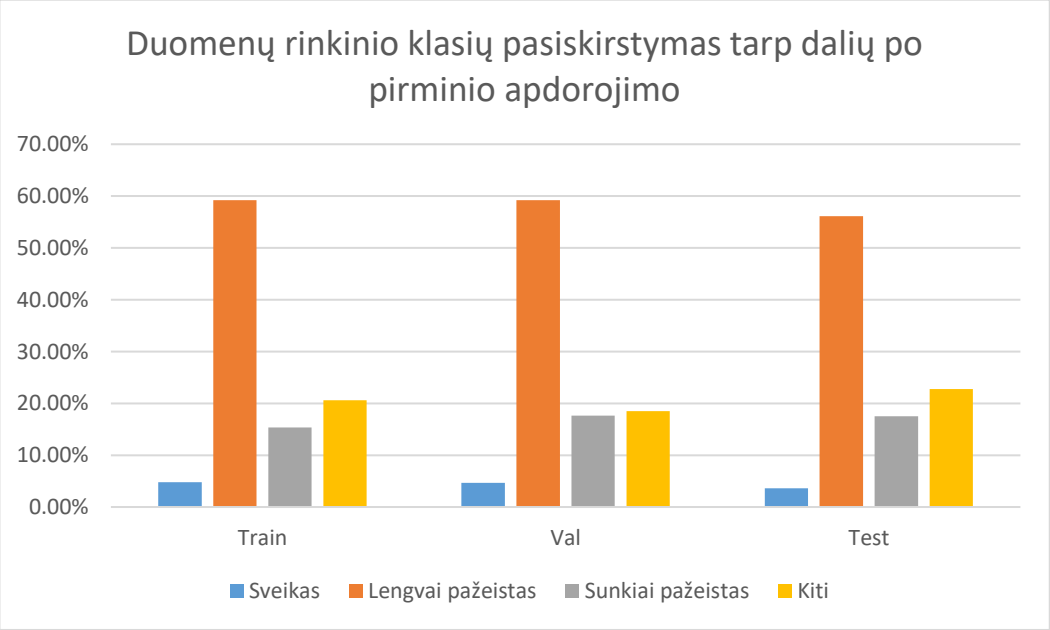
| **Klasė** | **Train** | **Val** | **Test** | **Iš viso** |
|---|---|---|---|---|
| Sveikas | 11 130 | 555 | 185 | 11 870 |
| Lengvai pažeistas | 137 076 | 7 019 | 2 881 | 146 976 |
| Sunkiai pažeistas | 35 563 | 2 093 | 899 | 38 555 |
| Kiti | 47 760 | 2 191 | 1 167 | 51 118 |
| Iš viso | 231 529 | 11 858 | 5 132 | 248 519 |

Objektų pasiskirstymas pagal klases po duomenų apdorojimo pateikiamas grafiškai stulpeline diagrama (žr. 29 pav.).




**29 pav.** Duomenų rinkinio klasių pasiskirstymas po pirminio apdorojimo

Kadangi naudojant mažą duomenų rinkinį duomenys buvo augmentuojami, prieš apmokant objektų aptikimo modelius siekiama įsitikinti, kad visuose mokymo, validavimo ir testavimo duomenų suskaidymuose, klasifikuojamų objektų pasiskirstymas yra panašus (žr. 30 pav.), t.y. nėra duomenų dalies kurioje nebūtų klasifikavimo pavyzdžių (ypač aktualu sveikų medžių daliai, nes šių duomenų pavyzdžių yra mažai).




**30 pav.** Duomenų rinkinio klasių pasiskirstymas tarp dalių po pirminio apdorojimo

Iš grafiko matyti, kad klasių proporcijos tarp mokymo, validavimo ir testavimo aibių išlieka panašios, todėl galima teigti, kad duomenų augmentacija ir padalijimas tinkamai išsaugo originalią klasių struktūrą ir leidžia patikimai vertinti modelių rezultatus.

## 3.3. Eksperimento sąlygos ir vertinimo metrikos

### 3.3.1. Hiperparametrai ir mokymo aplinka

Dauguma išbandytų modelių turi patogias apmokymo aplinkas, pavyzdžiui, YOLO ir RT-DETR modeliai buvo naudojami iš Ultralytics bibliotekos. Ši biblioteka yra gerai išvystyta ir naudojant automatinius mokymo nustatymus dažniausiai yra gaunami geriausi rezultatai. Vieninteliai hiperparametrai, kurie buvo keičiami šių modelių mokymo metu, yra vaizdo rezoliucija ir partijos dydis (angl. *batch size*). Visi YOLO ir RT-DETR modeliai buvo apmokyti su 512×512, 640×640 ir 1280×1280 pikselių rezoliucijomis. Šiame eksperimente vaizdo rezoliucija yra svarbus parametras, nes didelės raiškos aerovaizduose vienas medis gali užimti sąlyginai mažą vaizdo dalį. Modeliai apmokyti mažesne įvesties rezoliucija gali prarasti smulkius vizualinius požymius, svarbius pažeidimo lygio nustatymui, pavyzdžiui, lapų atspalvio variacijas arba lajos struktūros pokyčius. Kitų modelių – Faster R-CNN ir RF-DETR – apmokymui buvo taikomi modelių kūrėjų rekomenduojami hiperparametrai. Visi objektų aptikimo modeliai buvo mokomi 100 epochų.

Eksperimentinis tyrimas atliktas vietinėje darbo stotyje, kurios aparatinę įrangą sudarė Intel Core i7-14700K procesorius, 32 GB DDR5 6000 MHz operatyvioji atmintis, NVIDIA GeForce RTX 4060 Ti 16 GB vaizdo plokštė ir Samsung 990 PRO 2 TB NVMe kietasis diskas. Modelių apmokymas ir testavimas vykdytas CUDA skaičiavimų platformoje.

### 3.3.2. Vertinimo metrikos

Tyrimo metu apmokytų modelių veikimui vertinti naudojamos šios objektų aptikimo ir klasifikavimo metrikos:

- *mAP50* – vidutinis objektų aptikimo tikslumas, skaičiuojamas kai objekto ribas nustatančio stačiakampio ir tikrosios objekto vietos persidengimas viršija 50 %. Skaičiuojama nuo 0 iki 1 – didesnė reikšmė rodo tikslesnį objektų lokalizavimą ir objektų atpažinimą;
- *mAP50-95* – griežtesnė vidutinio tikslumo metrika, apskaičiuojama naudojant kelis skirtingus slenksčius nuo 50 % iki 95 % kas 5 % intervalą. Geresnis šios metrikos įvertis reiškia tikslesnį objektų lokalizavimą;
- tikslumas (angl. *precision*) – matuoja, kokia dalis modelio aptiktų objektų yra teisingi;
- atkūrimas (angl. *recall*) – matuoja, kokia dalis tikrų objektų yra aptikti modelio;
- *F1* balas – išvestinė metrika skaičiuojama vertinant tikslumą ir atkūrimą. Balansuoja tikslumą ir atkūrimą vienoje metrikoje. Didesnė reikšmė rodo geresnį modelio stabilumą, ypač esant netolygiam klasių pasiskirstymui;
- makro F1 – klasifikavimo įvertinimo metrika kai kiekvienai klasei paskaičiuojamas F1 balas atskirai ir išvedamas visų klasių aritmetinis vidurkis;
- vykdymo laikas (angl. *inference*) – vidutinis laikas vienam vaizdui apdoroti nuo įvesties iki prognozės skaičiuojamas milisekundėmis.

Šios metrikos buvo pasirinktos siekiant įvertinti kompiuterinės regos modelių veikimo kokybę vertinant tris pagrindinius aspektus. Vidutinis objektų aptikimo tikslumas leidžia įvertinti lokalizavimo kokybę – kaip tiksliai modelis aptinka ir lokalizuoja objektus vaizde. Tikslumas, atkūrimas ir F1 balo metrikos leidžia įvertinti klasifikavimo subalansuotumą, tai yra ypač svarbu esant netolygiam klasių pasiskirstymui. Vykdymo laikas leidžia palyginti nagrinėjamų

architektūrų apdorojimo greitį, tai yra svarbu vertinant modelių pritaikomumą realioms sąlygoms, kai skaičiavimo ištekliai gali būti riboti.

## 3.4. Rezultatai ir analizė

### 3.4.1. Vienos stadijos modelių rezultatai

Vienos stadijos strategijoje naudojami kompiuterinės regos modeliai kartu aptinka ir klasifikuoja medžių pažeidimus į 4 klases. Visų vienos stadijos modelių apmokymo rezultatai su skirtingomis hiperparametrų konfigūracijomis, įvertinti naudojant testavimo duomenų rinkinio dalį pateikiami 43 lentelėje.

**43 lentelė** Vienos stadijos modelių geriausi rezultatai

| Modelis | Vaizdo dydis | Partijos dydis | mAP50 | mAP50-95 | Tikslumas | Atkūrimas | F1 |
|---|---|---|---|---|---|---|---|
| YOLOv8m | 512 | 32 | 0,800 | 0,516 | 0,711 | 0,771 | 0,739 |
| YOLOv8m | 640 | 16 | 0,793 | 0,519 | 0,730 | 0,774 | 0,751 |
| YOLOv8m | 1280 | 8 | 0,804 | 0,536 | 0,732 | 0,768 | 0,750 |
| YOLOv9m | 512 | 32 | 0,794 | 0,519 | 0,724 | 0,776 | 0,749 |
| YOLOv9m | 640 | 16 | 0,806 | 0,535 | 0,724 | 0,786 | 0,754 |
| YOLOv9m | 1280 | 8 | 0,814 | 0,545 | 0,722 | 0,777 | 0,748 |
| YOLOv10m | 512 | 32 | 0,783 | 0,510 | 0,711 | 0,753 | 0,732 |
| YOLOv10m | 640 | 16 | 0,792 | 0,522 | 0,721 | 0,773 | 0,746 |
| YOLOv10m | 1280 | 8 | 0,809 | 0,544 | 0,755 | 0,758 | 0,756 |
| YOLOv11m | 512 | 32 | 0,792 | 0,515 | 0,697 | 0,792 | 0,741 |
| YOLOv11m | 640 | 16 | 0,782 | 0,513 | 0,701 | 0,781 | 0,739 |
| YOLOv11m | 1280 | 8 | 0,816 | 0,551 | 0,744 | 0,774 | 0,759 |
| YOLOv12m | 512 | 32 | 0,781 | 0,507 | 0,689 | 0,766 | 0,725 |
| YOLOv12m | 640 | 16 | 0,801 | 0,526 | 0,698 | 0,784 | 0,739 |
| YOLOv12m | 1280 | 8 | 0,809 | 0,550 | 0,741 | 0,779 | 0,760 |
| RT-DETR | 512 | 32 | 0,698 | 0,460 | 0,700 | 0,737 | 0,718 |
| RT-DETR | 640 | 16 | 0,692 | 0,462 | 0,710 | 0,724 | 0,717 |
| RT-DETR | 1280 | 8 | 0,698 | 0,474 | 0,709 | 0,711 | 0,710 |
| Faster R-CNN | – | 2 | 0,704 | 0,410 | 0,647 | 0,724 | 0,681 |
| RF-DETR | 560 | 4 (×4) | 0,747 | 0,484 | 0,833 | 0,772 | 0,801 |

Rezultatų palyginamojoje lentelėje matoma, kad objektų aptikimo užduotį geriausiai atlieka YOLO modeliai. Geriausią rezultatą pasiekė YOLOv11m modelis, apmokytas naudojant 1280×1280 raiškos vaizdų įvesties rezoliuciją. Geriausius rezultatus pasiekusio modelio vidutinis objektų aptikimo tikslumas yra 0,816 ir mAP50–95 tikslumas 0,551. Skirtumai tarp geriausiai pasirodžiusių YOLO modelių yra labai nedideli, jų veikimas yra panašaus patikimumo. Tačiau RF-DETR modelis išsiskiria aukščiausia tikslumo ir F1 balo reikšmėmis: tikslumas 0,833; F1 balas 0,801. Tačiau RF-DETR modelis pagal bendrą objektų lokalizavimo kokybę nusileidžia YOLO modeliams: vidutinis objektų aptikimo tikslumas 0,747; mAP50–95 tikslumas 0,484. Taigi tyrimo kontekste YOLO architektūra geriausiai

tinka medžių aptikimo užduočiai, o RF-DETR pasiekė geresnius rezultatus klasifikavimo patikimumo metrikose. RT-DETR ir Faster R-CNN modeliai visomis metrikomis nusileidžia YOLO ir RF-DETR modeliams, todėl šiame tyrime nėra konkurencingi ir tolesniuose tyrimo etapuose nenaudojami.

Įvertinus visus rezultatus pastebima, kad nėra vieno modelio, kuris išsiskirtų ir pasiektų geriausius rezultatus visose metrikose – YOLO ir RF-DETR pasižymi savitais privalumais ir trūkumais priklausomai nuo vertinamos metrikos. Rezultatai taip pat rodo, kad įvesties vaizdo rezoliucija turi teigiamą įtaką YOLO modelių aptikimo kokybei. Padidinus rezoliuciją nuo 512×512 iki 1280×1280 pikselių, vidutinis objektų aptikimo tikslumas mAP50 vidutiniškai padidėja apie 2-3 %, o mAP50-95 apie 3-4 %. Didesnės raiškos vaizduose modelis geriau atskiria smulkius objektus ir tiksliau nustato jų ribas.

### 3.4.2. Dviejų stadijų modelių rezultatai

#### 3.4.2.1. Objektų aptikimas

Dviejų stadijų požiūryje pirmojoje stadijoje atliekamas medžių aptikimas, sujungiant visas keturias pažeidimų klases į vieną bendrą klasę medis, o antrojoje stadijoje klasifikavimo modelis kiekvienam aptiktam objektui priskiria konkrečią medžio lajos pažeidimo klasę. Pirmosios stadijos objektų aptikimo modeliu pasirinktas YOLOv11m, kuris vienos stadijos eksperimentuose pasiekė geriausius vidutinio objektų aptikimo tikslumo mAP50 rezultatus. Modelis buvo iš naujo apmokytas naudojant tą patį duomenų rinkinį, tačiau jį pritaikius tik vienos klasės aptikimo užduočiai. Siekiant užtikrinti rezultatų palyginamumą, buvo panaudota ta pati hiperparametrų konfigūracija – 1280×1280 pikselių įvesties rezoliucija. Gauti rezultatai pateikiami 44 lentelėje.

**44 lentelė** Pirmosios stadijos aptikimo rezultatai naudojant YOLOv11m

| Metrika | YOLOv11m |
|---|---|
| mAP50 | 0,965 |
| mAP50-95 | 0,666 |
| Tikslumas | 0,919 |
| Atkūrimas | 0,913 |

Lentelėje pateikti rezultatai rodo, kad medžių aptikimas vienos klasės užduotyje pasiekia ženkliai aukštesnį tikslumą nei keturių klasių aptikimo atvejais: vidutinis objektų aptikimo tikslumas mAP50 0,965; mAP50-95 tikslumas 0,666; tikslumas 0,919; atkūrimas 0,913. Aukštesnius vienos klasės užduoties rezultatus lemia tai, kad medžio buvimo nustatymas yra žymiai paprastesnė užduotis nei konkretaus pažeidimo lygio klasifikavimas. Aukštos ir subalansuotos tikslumo bei atkūrimo reikšmės leidžia teigti, kad modelis vienodai gerai išvengia tiek praleistų, tiek klaidingų aptikimų. Vienos klasės modelio mAP50–95 taip pat pranoksta geriausio keturių klasių modelio rezultatą, o tai rodo pakankamą ribojančių stačiakampių tikslumą antrosios stadijos apkarpymui. Klasifikatoriui svarbu, kad medis patektų į apkarpytą sritį, todėl absoliutus stačiakampio ribų tikslumas nėra kritinis.

Šie rezultatai sudaro tvirtą pagrindą antrajai klasifikavimo stadijai. Tačiau svarbu atkreipti dėmesį, kad net 0,965 mAP50 tikslumo rezultatas reiškia, kad modelis vis dar gali dalį

objektų aptikti netiksliai arba jų visai neaptikti. Šie atvejai tampa neištaisomomis klaidomis dviejų stadijų sistemoje, nes klasifikatorius apdoroja tik pirmojoje stadijoje aptiktus objektus.

#### 3.4.2.2. Objektų klasifikavimas

Antrojoje stadijoje aptiktos medžių sritys yra apkarpomos pagal pirmojoje stadijoje nustatytų objektų ribojančius stačiakampius, normalizuojamos iki 224×224 raiškos ir pateikiamos vienam iš trijų tyrime naudojamų klasifikavimo modelių. Klasifikatoriai apmokyti naudojant tuos pačius mokymo, validavimo ir testavimo duomenų padalijimus kaip ir aptikimo modeliai, tačiau jų įvestį sudaro iškirpti medžių vaizdai, o ne pilnos aeronuotraukos. Apmokant objektų klasifikavimo modelius yra taikoma augmentacija, horizontalus apvertimas ir maži vaizdo pasukimai, siekiant pagerinti klasifikatoriaus apibendrinimo gebą. Visų trijų klasifikavimo modelių rezultatai pateikiami 45 lentelėje.

**45 lentelė** Klasifikavimo modelių rezultatai

| Metrika | ResNet101 | ConvNeXtV2 | EfficientNet V2-S |
|---|---|---|---|
| Sveikas F1 | 0,5605 | 0,5311 | 0,5868 |
| Lengvai pažeistas F1 | 0,9036 | 0,9046 | 0,9061 |
| Sunkiai pažeistas F1 | 0,7914 | 0,7925 | 0,7949 |
| Kiti F1 | 0,9850 | 0,9803 | 0,9860 |
| Makro F1 vidurkis | 0,8102 | 0,8021 | 0,8185 |

Klasifikavimo modelių rezultatų palyginamojoje lentelėje matoma, kad visi trys modeliai pasirodė labai panašiai – nė vienas reikšmingai nepralenkia kitų. Geriausius rezultatus pasiekė EfficientNet V2-S modelis: sveikų medžių klasės F1 balas 0,5868; lengvai pažeistų 0,9061; sunkiai pažeistų 0,7949; kitų medžių 0,9860; makro F1 vidurkis 0,8185. Prasčiau pasirodžiusių modelių makro F1 vidurkiai taip pat yra aukšti: ResNet101 siekia 0,8102, o ConvNeXtV2 – 0,8021, taigi skirtumai tarp visų trijų modelių yra minimalūs. Visi trys modeliai prasčiausiai klasifikuoja sveikų medžių klasę, tai tikėtina susiję su šios klasės duomenų trūkumu ir klasių disbalansu. Pažymėtina, kad geriausius rezultatus šioje klasėje pasiekė mažiausias modelis pagal parametrų skaičių – EfficientNet V2-S. Tai patvirtina šio modelio architektūros efektyvumo principą, pagal kurį subalansuotas tinklo dydžio mastelio keitimas leidžia pasiekti gerus rezultatus su mažesniais skaičiavimo ištekliais ir ribotu duomenų kiekiu.

### 3.4.3. Vienos stadijos ir dviejų stadijų architektūrų palyginimas

#### 3.4.3.1. Tikslumo analizė

Pagrindinis tyrimo tikslas – nustatyti, ar dviejų stadijų architektūra gali pasiekti geresnius medžių pažeidimų klasifikavimo rezultatus nei vienos stadijos modeliai. Šių architektūrų palyginimo rezultatai pateikiami 46 lentelėje, kurioje vienodomis vertinimo sąlygomis lyginamos geriausias tikslumo metrikas pasiekusios modelių konfigūracijos. Visi modeliai buvo testuojami naudojant tą patį, mokymo metu nematytą duomenų rinkinio testavimo dalį. Kiekvienas modelis įvertino vaizdus ir pateikė aptiktų medžių klases ir jų vietas vaizde. Gautos prognozės buvo palygintos su tikrosiomis sužymėjimų vertėmis, siekiant nustatyti, kurie aptikimai buvo teisingi, o kurie klaidingi arba praleisti. Modelių veikimas buvo

vertinamas apskaičiuojant F1 metrikas kiekvienai klasei bei bendrą makro F1 vidurkį. Toks vertinimo metodas leido vienodomis sąlygomis palyginti vienos ir dviejų stadijų architektūrų klasifikavimo tikslumą.

**46 lentelė** Geriausių vienos ir dviejų stadijų konfigūracijų tikslumo metrikų palyginimas

| **Metrika** | **Vienos stadijos, YOLOv11m** | **Vienos stadijos, RF-DETR** | **Dviejų stadijų, YOLOv11m + EfficientNet V2-S** |
|---|---|---|---|
| Sveikas F1 | 0,5337 | 0,5988 | 0,5977 |
| Lengvai pažeistas F1 | 0,8505 | 0,8810 | 0,8413 |
| Sunkiai pažeistas F1 | 0,7382 | 0,7058 | 0,7192 |
| Kiti F1 | 0,7483 | 0,7282 | 0,7343 |
| Makro F1 vidurkis | 0,7177 | 0,7284 | 0,7231 |

Rezultatų palyginamojoje lentelėje matoma, kad geriausią bendrą klasifikavimo rezultatą pasiekė vienos stadijos modelis RF-DETR, kurio makro F1 vidurkis yra 0,7284. Tačiau skirtumai nuo dviejų stadijų YOLOv11m + EfficientNet V2-S architektūros, kurios makro F1 vidurkis pasiekė 0,7231, ir vienos stadijos YOLOv11m modelio, kurio makro F1 vidurkis pasiekė 0,7177, yra minimalūs. Geriausią bendrą rezultatą pasiekęs RF-DETR modelis išsiskiria geriausiais rezultatais sveikų medžių klasėje, kurioje F1 balas siekia 0,5988, ir lengvai pažeistų medžių klasėje, kurioje F1 balas siekia 0,8810, tačiau nusileidžia YOLOv11m sunkiai pažeistų ir kitų medžių klasėse. Vienos stadijos YOLOv11m modelis pasiekė geriausius rezultatus sunkiai pažeistų medžių klasėje, kurioje F1 balas siekia 0,7382, ir kitų medžių klasėje, kurioje F1 balas siekia 0,7483. Dviejų stadijų YOLOv11m + EfficientNet V2-S architektūra pagerina bendrą klasifikavimo tikslumą, lyginant su vienos stadijos YOLOv11m modelio rezultatais, tačiau vis tiek nusileidžia RF-DETR modeliui ir nepasiekia pranašumo nei vienoje atskiroje klasėje. Dviejų stadijų architektūra kenčia nuo klaidų persidenginimo tarp stadijų, nes kiekvienas pirmojoje stadijoje praleistas medis tampa neatšaukiama klaida. Todėl nepaisant aukšto EfficientNet V2-S klasifikatoriaus tikslumo, kurio makro F1 vidurkis pasiekė 0,8185, galutinis dviejų stadijų architektūros rezultatas nėra geresnis nei vienos stadijos modelių.

Įvertinus visus rezultatus, galima teigti, kad dviejų stadijų metodas nesuteikia pranašumo prieš vienos stadijos sprendimus. Skirtumas tarp geriausio ir blogiausio modelio makro F1 vidurkio siekia tik 1,1 %. Taigi, lyginant šiuos modelius, galima teigti, kad nėra modelio, kurio rezultatai būtų reikšmingai geresni, visi trys sprendimai pasiekia panašų klasifikavimo tikslumą. Galima teigti, kad modelio pasirinkimą praktiniam taikymui gali lemti ne klasifikavimo tikslumas, o kiti veiksniai, pavyzdžiui, skaičiavimo resursų poreikis ar veikimo greitis.

#### 3.4.3.2. Apdorojimo laiko analizė

Vienos ir dviejų stadijų architektūrų apdorojimo laiko bei resursų sąnaudų palyginimas pateikiamas 47 lentelėje. Skaičiavimo resursų ir apdorojimo laiko vertinimas buvo atliktas naudojant 85 vaizdų turinčią duomenų rinkinio testavimo dalį, kurioje iš viso yra 5 216 objektų-medžių su skirtingais pažeidimų lygiais.

**47 lentelė** Vienos ir dviejų stadijų architektūrų skaičiavimo resursų palyginimas

| Modelis | Bendras apdorojimo laikas (s) | Apdorojimo laikas vienam vaizdui (ms) | Aptikimų skaičius |
|---|---|---|---|
| Vienos stadijos, YOLOv11m | 4,411 | 51,9 | 5 294 |
| Vienos stadijos, RF-DETR | 15,9 | 187 | 5 133 |
| Dviejų stadijų, YOLOv11m + ResNet101 | 17,713 | 208,4 | 5 646 |
| Dviejų stadijų, YOLOv11m + ConvNeXtV2 | 18,807 | 221,3 | 5 646 |
| Dviejų stadijų, YOLOv11m + EfficientNet V2-S | 14,941 | 175,8 | 5 646 |

Skaičiavimo resursų palyginamojoje lentelėje matoma, kad greičiausiu apdorojimo laiku išsiskiria vienos stadijos YOLOv11m modelis, kuris užtrunka 51,9 ms vienam vaizdui apdoroti. Tai atitinka šios modelio architektūros paskirtį – YOLO modeliai tradiciškai kuriami realaus laiko aptikimo užduotims, todėl greitis yra vienas pagrindinių jų projektavimo prioritetų. Geriausiai pasirodęs modelis, vienos stadijos RF-DETR, vieną vaizdą apdoroja per 187 ms, t.y. panašiu greičiu kaip ir greičiausia dviejų stadijų konfigūracija YOLOv11m su EfficientNet V2-S, kurios apdorojimo laikas siekia 175,8 ms. Tai reiškia, kad dviejų stadijų architektūra ne tik nusileidžia geriausiai pasirodžiusiam vienos stadijos RF-DETR modeliui klasifikavimo tikslumu, bet ir yra daug lėtesnis sprendimas vertinant apdorojimo greitį, todėl praktinio taikymo požiūriu ši architektūra nepateisina didesnio sistemos sudėtingumo.

## 3.5. Rezultatų aptarimas ir išvados

Lyginant vienos ir dviejų stadijų architektūras, paaiškėjo, kad bendras klasifikavimo tikslumas tarp jų skiriasi neženkliai. Dviejų stadijų YOLOv11m + EfficientNet V2-S sprendimas leido geriau suvaldyti klasių disbalansą, atskiras klasifikatorius su klasių svoriais ženkliai pagerino sveikų medžių klasės F1 balą, todėl neženkliai pagerino bendrą makro F1 vidurkį. Tačiau, šio pagerėjimo nepakako, kad dviejų stadijų metodas pranoktų geriausią vienos stadijos RF-DETR modelį pagal bendrą makro F1 vidurkį. Be to, dviejų stadijų architektūra yra vidutiniškai 4 kartus lėtesnė už vienos stadijos YOLOv11m modelį, todėl didesnio sistemos sudėtingumo ir sulėtėjimo nepateisina nei tikslumas, nei greitis. Taigi vienos stadijos transformeriais grįstas RF-DETR sprendimas tyrime pasirodė geriausiai tikslumo rezultatu ir greičiu tik nežymiai nusileido sparčiausiai dviejų stadijų architektūrai.

Klasių disbalansas išliko esminiu iššūkiu viso tyrimo metu: sveikų medžių klasė sudarė tik 4,5 % duomenų, todėl visų nagrinėtų modelių rezultatai šioje klasėje buvo reikšmingai prastesni nei kitose klasėse. Pažymėtina, kad esant didesniam sveikų medžių klasės duomenų kiekiui ir tolygesniam klasių pasiskirstymui, vienos stadijos YOLO modeliai tikėtinai galėtų pasiekti panašius rezultatus be papildomo architektūrinio sudėtingumo.

Apibendrinant tyrimo rezultatus, galima teigti, kad dviejų stadijų architektūra yra nepasiteisinantis sprendimas  geresnis klasifikavimo tikslumas (pasiekiamas antroje stadijoje) nekompensuoja pirmoje stadijoje padarytų klaidų, kur kiekvienas praleistas, neaptiktas medis tampa neatšaukiama klaida. Dėl šios priežasties bendras klasifikavimo tikslumas yra prastesnis ir ženkliai nepagerina bendro klasifikavimo tikslumo lyginant su vienos stadijos modeliais. Todėl, galima teigti, kad vienos stadijos modeliai yra optimalesnis sprendimas nei dviejų stadijų architektūra.

Lyginant vienos stadijos modelius, RF-DETR pasiekė geresnį bendrą klasifikavimo rezultatą už YOLOv11m, tačiau analizuojant atskirų klasių lygmeniu, abiejų modelių rezultatai buvo panašūs – abu modeliai pasiekė geriausius rezultatus dvejose klasėse iš keturių. Vertinant apdorojimo greitį, YOLOv11m yra apie tris kartus spartesnis už RF-DETR modelį. Abu modeliai yra konkurencingi praktiniam pritaikymui. Dėl sparčiausio apdorojimo laiko ir konkurencingo klasifikavimo tikslumo YOLOv11m modelis yra optimalus pasirinkimas daugumai praktinių miško monitoringo taikymų. RF-DETR modelis geriau atpažino retas pažeidimų klases ir pasižymėjo didesniu bendru klasifikavimo tikslumu.

## Išvados

1. Atlikus literatūros analizę nustatyta, kad tradicinis medžių defoliacijos vertinimas yra brangus, subjektyvus ir reikalauja didelių žmogiškųjų išteklių, todėl automatizuotų sprendimų taikymas gali padėti greičiau, objektyviau ir efektyviau atlikti miško būklės stebėjimą ir taupyti dalykinės srities ekspertų laiką.
2. Atlikus literatūros analizę ir panašių sprendimų apžvalgą, buvo susipažinta su populiariausiomis kompiuterinės regos ir giluminio mokymosi architektūromis, taikomomis vaizdų klasifikavimo bei objektų aptikimo uždaviniuose. Išanalizuotos konvoliucinių neuroninių tinklų, vienos stadijos detektorių ir transformerių pagrindu sukurtų modelių architektūros bei jų taikymo ypatumai. Nustatyta, kad modelių veikimo kokybei didelę įtaką daro mokymo duomenų kiekis, jų kokybė, klasių balansas ir duomenų įvairovė.
3. Duomenų rinkinio analizė parodė, kad pasirinktas duomenų rinkinys pasižymi klasių disbalansu – kai kurios kategorijos turėjo ženkliai daugiau pavyzdžių nei kitos. Nustatyta, kad toks netolygus duomenų pasiskirstymas neigiamai veikia modelių mokymosi procesą ir mažina prasčiau atstovaujamų klasių klasifikavimo tikslumą.
4. Eksperimentų metu buvo apmokyti ir palyginti vienos bei dviejų stadijų architektūrų modeliai, skirti miško medžių lajų pažeidimų klasifikavimui iš UAV vaizdų. Tyrimo rezultatai parodė, kad vienos stadijos modeliai pasiekė panašų arba geresnį klasifikavimo tikslumą nei dviejų stadijų architektūra, kartu pasižymėdami didesniu apdorojimo greičiu ir mažesniu sistemos sudėtingumu. Nustatyta, kad dviejų stadijų sprendimuose pirmojo etapo objektų aptikimo klaidos tiesiogiai riboja galutinį klasifikavimo rezultatą.
5. Eksperimentų metu pastebėta, kad įvesties vaizdų rezoliucija turi mažą įtaką objektų aptikimo kokybei – didesnė rezoliucija leidžia geriau išsaugoti smulkius medžių lajų požymius ir pagerina modelių tikslumą analizuojant aerovaizdus.
6. Tyrimo metu nustatyta, kad klasių disbalansas išliko vienu svarbiausių veiksnių, ribojančių modelių veikimo kokybę. Mažiausiai duomenų turinti sveikų medžių klasė buvo klasifikuojama prasčiausiai, todėl tolygesnis duomenų pasiskirstymas tikėtinai leistų reikšmingai pagerinti bendrus modelių rezultatus.
7. Atliktas tyrimas parodė, kad vienos stadijos objektų aptikimo modeliai yra tinkamas ir praktiškai pritaikomas sprendimas automatiniam miško medžių lajų pažeidimų klasifikavimui iš UAV vaizdų. Tyrimo rezultatai atskleidė, kad skirtumai tarp geriausiai pasirodžiusių modelių buvo minimalūs. YOLO ir transformerių pagrindu sukurti RF-DETR modeliai pasiekė labai panašius klasifikavimo rezultatus. Tačiau, YOLO modeliai užduotį atliko tris kartus greičiau nei RF-DETR ar dviejų stadijų architektūros sprendimai, todėl dėl savo efektyvumo ir spartaus veikimo išlieka konkurencingas pasirinkimas praktiniuose taikymuose ir labiau tinka realaus laiko ar spartaus apdorojimo sistemoms, o transformerių pagrindu sukurti modeliai gali būti naudingi, kai svarbiausias yra didesnis klasifikavimo tikslumas.

## Literatūros sąrašas


[1] V. Stakėnas, I. Varnagirytė-Kabašinskienė, ir A. Kabašinskas, „A methodological approach for the assessment of basic crown parameters in scots pine stands“, *Balt. For.*, t. 27, nr. 1, 2021, doi: 10.46490/BF551.

[2] A. Michel ir W. Seidling, „Forest Condition in Europe 2014 technical report of ICP forests ; report under the UNECE Convention on Long-Range Transboundary Air Pollution (CLRTAP)“, Vienna, 2014.

[3] Čapkauskas G., „Gamtiniai medynų vystymosi trūkumai: medžių pažeidžiamumas ir lajų defoliacijos rizika.“, Aleksandro Stulginskio universitetas, Kaunas, 2016.

[4] Federal Research Centre for Forestry and Forest Products (BFH), „Convention on Long-Range Transboundary Air Pollution, International Co-operative Programme on Assessment and Monitoring of Air Pollution Effects on Forests, and European Union Scheme on the Protection of Forests Against Atmospheric Pollution prepared by: Federal Research Centre for Forestry and Forest Products (BFH)“, Geneva and Brussels, 2003.

[5] S. Solberg ir L. Strand, „Crown density assessments, control surveys and reproducibility“, 1999. doi: 10.1023/A:1005980326079.

[6] K. Budnik, J. Byrtek, B. Skrabanek, ir J. Wajs, „AI-Accelerated Decision Making in Forest Management“, *IOP Conference Series: Earth and Environmental Science*, Institute of Physics, 2023. doi: 10.1088/1755-1315/1189/1/012030.

[7] N. Mizoue, „CROCO : Semi-automatic Image Analysis System fbr Crown Condition Assessment in Forest Health Monitoring“, 2022. doi: 10.20659/jfp.8.1_17.

[8] U. Kälin, N. Lang, C. Hug, A. Gessler, ir J. D. Wegner, „Defoliation estimation of forest trees from ground-level images“, 2018, doi: 10.1101/441733.

[9] „Pl@ntNet“. Žiūrėta: 2023 m. gruodžio 11 d. [Interaktyvus]. Adresas: https://identify.plantnet.org/

[10] „LeafSnap“. Žiūrėta: 2023 m. gruodžio 11 d. [Interaktyvus]. Adresas: https://plantidentifier.info/

[11] „iNaturalist“. Žiūrėta: 2023 m. gruodžio 11 d. [Interaktyvus]. Adresas: https://www.inaturalist.org/

[12] B. Ho, B. B. Kocer, ir M. Kovac, „Vision based crown loss estimation for individual trees with remote aerial robots“, *ISPRS Journal of Photogrammetry and Remote Sensing*, t. 188, p. 75–88, birž. 2022, doi: 10.1016/j.isprsjprs.2022.04.002.

[13] Y. Zhang, D. Zhou, S. Chen, S. Gao, ir Y. Ma, „Single-Image Crowd Counting via Multi-Column Convolutional Neural Network“, Las Vegas: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. doi: 10.1109/CVPR.2016.70.

[14] Y. Chen, J. Yang, B. Chen, ir S. Du, „Counting Varying Density Crowds Through Density Guided Adaptive Selection CNN and Transformer Estimation“, birž. 2022, doi: 10.1109/TCSVT.2022.3208714.

[15] L. Dong, H. Zhang, Y. Ji, ir Y. Ding, „Crowd counting by using multi-level density-based spatial information: A Multi-scale CNN framework“, *Inf. Sci. (N. Y).*, t. 528, p. 79–91, rugpj. 2020, doi: 10.1016/j.ins.2020.04.001.

[16] M. R. Bhuiyan *ir kt.*, „A deep crowd density classification model for Hajj pilgrimage using fully convolutional neural network“, *PeerJ Comput. Sci.*, t. 8, 2022, doi: 10.7717/peerj-cs.895.
[17] X. Sun, L. Xu, Y. Zhou, ir Y. Shi, „Leaves and Twigs Image Recognition Based on Deep Learning and Combined Classifier Algorithms“, *Forests*, t. 14, nr. 6, birž. 2023, doi: 10.3390/f14061083.
[18] S. Liu ir R. Vetter, „Enhancing Training Data Quality With Visual Analytics“, *Computer (Long. Beach. Calif).*, t. 56, nr. 11, p. 4–6, lapkr. 2023, doi: 10.1109/MC.2023.3304568.
[19] C.-C. J. Kuo, „Understanding Convolutional Neural Networks with A Mathematical Model“, lapkr. 2016, [Interaktyvus]. Adresas: http://arxiv.org/abs/1609.04112
[20] S. Khan, H. Rahmani, S. A. A. Shah, ir M. Bennamoun, „Convolutional Neural Network“, *A Guide to Convolutional Neural Networks for Computer Vision*, S. Khan, H. Rahmani, S. A. A. Shah, ir M. Bennamoun, Sud., Cham: Springer International Publishing, 2018, p. 43–68. doi: 10.1007/978-3-031-01821-3_4.
[21] R. Yamashita, M. Nishio, R. K. G. Do, ir K. Togashi, „Convolutional neural networks: an overview and application in radiology“, 2018 m. rugpjūčio 1 d., *Springer Verlag*. doi: 10.1007/s13244-018-0639-9.
[22] K. He, X. Zhang, S. Ren, ir J. Sun, „Deep Residual Learning for Image Recognition“, gruodž. 2015, [Interaktyvus]. Adresas: http://arxiv.org/abs/1512.03385
[23] M. Tan ir Q. V. Le, „EfficientNetV2: Smaller Models and Faster Training“, birž. 2021, [Interaktyvus]. Adresas: http://arxiv.org/abs/2104.00298
[24] S. Woo *ir kt.*, „ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders“, saus. 2023, [Interaktyvus]. Adresas: http://arxiv.org/abs/2301.00808
[25] W. Luo, Y. Li, R. Urtasun, ir R. Zemel, „Understanding the Effective Receptive Field in Deep Convolutional Neural Networks“, saus. 2017, [Interaktyvus]. Adresas: http://arxiv.org/abs/1701.04128
[26] A. Dosovitskiy *ir kt.*, „An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale“, birž. 2021, [Interaktyvus]. Adresas: http://arxiv.org/abs/2010.11929
[27] S. Ren, K. He, R. Girshick, ir J. Sun, „Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks“, saus. 2016, [Interaktyvus]. Adresas: http://arxiv.org/abs/1506.01497
[28] J. Terven ir D. Cordova-Esparza, „A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS“, vas. 2024, doi: 10.3390/make5040083.
[29] A. Vaswani *ir kt.*, „Attention Is All You Need“, 2023.
[30] Y. Zhao *ir kt.*, „DETRs Beat YOLOs on Real-time Object Detection“, bal. 2024, [Interaktyvus]. Adresas: http://arxiv.org/abs/2304.08069
[31] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, ir J. Dai, „Deformable DETR: Deformable Transformers for End-to-End Object Detection“, kovo 2021, [Interaktyvus]. Adresas: http://arxiv.org/abs/2010.04159
[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, ir L. Fei-Fei, „ImageNet: A large-scale hierarchical image database“, *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, p. 248–255. doi: 10.1109/CVPR.2009.5206848.

[33] „Forest Damages – Larch Casebearer“. Žiūrėta: 2026 m. gegužės 4 d. [Interaktyvus]. Adresas: https://lila.science/datasets/forest-damages-larch-casebearer/