Kauno technologijos universitetas

Informatikos fakultetas

Kelyje matomo vaizdo segmentavimo sistema

Baigiamasis magistro studijų projektas

Bartas Lisauskas

Projekto autorius

Prof. dr. Rytis Maskeliūnas

Vadovas

Kaunas, 2025

Kauno technologijos universitetas

Informatikos fakultetas

Kelyje matomo vaizdo segmentavimo sistema

Baigiamasis magistro studijų projektas

Programų sistemų inžinerija (6211BX011)

Bartas Lisauskas

Projekto autorius

Prof. dr. Rytis Maskeliūnas

Vadovas

Prof. dr. Tomas Blažauskas

Recenzentas

Kaunas, 2025

Kauno technologijos universitetas

Informatikos fakultetas

Bartas Lisauskas

Kelyje matomo vaizdo segmentavimo sistema

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

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

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

3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs;

4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalintas iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Bartas Lisauskas

Patvirtinta elektroniniu būdu

Kauno technologijos universitetas

Informatikos fakultetas

Baigiamojo magistro projekto užduotis

Projekto tema: Kelyje matomo vaizdo segmentavimo sistema

Reikalavimai ir sąlygos (tikslinti pavadinimą pagal poreikį):

Vadovas / Vadovė: Prof. dr. Rytis Maskeliūnas

(vadovo pareigos, vardas, pavardė, parašas) (data)

Lisauskas Bartas. Kelyje matomo vaizdo segmentavimo sistema. Magistro studijų baigiamasis projektas, vadovas Prof. dr. Rytis Maskeliūnas; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Programų sistemų inžinerija.

Reikšminiai žodžiai: kompiuterinė rega, gilusis mokymasis, vaizdo apdorojimas, neuroniniai tinklai, semantinis segmentavimas.

Kaunas, 2025. 86 p.

Santrauka

Tikslus objektų ir aplinkos aptikimas yra vienas iš esminių reikalavimų norint taikyti kompiuterinės regos sistemas automobilių pramonės, robotikos ar aviacijos srityse. Kompiuterinės regos sistemų kokybė daro stiprią įtaką autonominių transporto priemonių veiksmams kelyje. Autonominėms transporto priemonėms siekiant išvengti eismo įvykių, yra būtina tiksliai identifikuoti kitus eismo dalyvius ar kliūtis kelyje. Dėl daugybės kelyje esančių objektų ir aplinkos įvairovės vaizdo segmentavimas išlieka viena iš sudėtingiausių užduočių kompiuterinės regos srityje.

Šiame darbe nagrinėjami kompiuterinės regos vaizdo segmentavimo modeliai, jų architektūros ir duomenų rinkiniai, siekiant sukurti efektyvią kelyje matomo vaizdo segmentavimo sistemą. Remiantis efektyviausių kompiuterinės regos modelių analizę, realizuojamas vaizdo segmentavimo algoritmas paremtas transformatoriaus neuroninio tinklo architektūra. Eksperimentinėje dalyje aprašomos ir palyginamos atliktos sistemos modifikacijos ir jų įtaka vaizdo segmentavimo sistemos tikslumui.

Lisauskas Bartas. Road Scene Segmentation System. Master's Final Degree Project supervisor Prof. dr. Rytis Maskeliūnas; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Software Engineering.

Keywords: computer vision, deep learning, image processing, neural networks, semantic segmentation.

Kaunas, 2025. 86 p.

Summary

The precise detection of objects and the surrounding environment is one of the essential requirements for applying computer vision systems in the automotive, robotic, and aviation industry. The quality of such systems has a strong influence on the actions of autonomous vehicles on the road. To avoid traffic accidents, autonomous vehicles must accurately identify other road users and obstacles. Due to the wide variety of objects and environmental conditions encountered on the road, image segmentation remains one of the most challenging tasks in the field of computer vision.

This thesis examines computer vision models for image segmentation, their architectures, and the data sets used, with the aim of creating an efficient system to segment road scene imagery. Based on an analysis of the most effective existing models, an image segmentation algorithm built upon a transformer-based neural network architecture is implemented. In the experimental section, modifications applied to the system are described and compared, and their impact on the accuracy of the image segmentation system is assessed.

Turinys

Lentelių sąrašas

Paveikslų sąrašas

Santrumpų ir terminų sąrašas

Santrumpos:

mIoU (angl. Mean Intersection over Union) – vidutinė persidengimo metrika;

ReLU (angl. Rectified Linear Unit) – aktyvacijos funkcija;

UML (angl. Unified Modeling Language) – modeliavimo ir specifikacijų kūrimo kalba.

Terminai:

Gilioji architektūra – neuroninio tinklo architektūra, turinti daug sluoksnių, dažnai naudojama sudėtingoms problemoms spręsti, pavyzdžiui, vaizdui atpažinti ar natūraliajai kalbai apdoroti.

Gilusis mokymasis – mašininio mokymosi sritis, kurioje naudojami neuroniniai tinklai ir kiti algoritmai, siekiant išmokti sudėtingų duomenų reprezentacijų bei atlikti tikslias prognozes ar klasifikaciją.

Konvoliucija – matematinė operacija, naudojama konvoliuciniuose neuroniniuose tinkluose, taikant įvairius filtrus įvesties duomenims, siekiant išskirti reikšmingas vaizdo savybes.

Konvoliucinis neuroninis tinklas – dirbtinio neuroninio tinklo rūšis, dažnai taikoma kompiuterinės regos užduotims atlikti, tokioms kaip objektų atpažinimas, klasifikacija ir segmentavimas.

Mašininis mokymasis – dirbtinio intelekto sritis, suteikianti kompiuteriams galimybę automatiškai mokytis iš duomenų ir atlikti prognozes ar sprendimus, minimizuojant žmogaus intervenciją.

Mokomasis vektorius – duomenų ar savybių masyvas, naudojamas modelio apmokymui ir prognozėms.

Perceptronas – dirbtinio neuroninio tinklo elementas, atliekantis skaičiavimus pagal įvestis ir svorius.

Transformeris – dirbtinio neuroninio tinklo architektūros tipas, paremtas dėmesio mechanizmu, dažnai naudojamas vaizdui atpažinti ar natūraliajai kalbai apdoroti.

Vaizdo segmentavimas – procesas, kurio metu vaizdas suskirstomas į skirtingas dalis, išskiriant skaitmeniniame vaizde esančius objektus ar reikšmingas sritis.

Įvadas

Kompiuterinė rega yra dirbtinio intelekto sritis, kuri moko kompiuterius suprasti žmonių matomą pasaulį. Neuroniniai tinklai yra taikomi objektų aptikimo bei klasifikavimo užduotims atlikti. Objektų ir aplinkos aptikimas vis labiau taikomas automobilių pramonėje, aviacijoje ir robotikoje. Naujausios kompiuterinės regos sistemos gali atpažinti vaizdinius duomenis greičiau ir tiksliau nei žmonės. Šių sistemų panaudojimas padeda automatizuoti procesus įvairiose srityse. Inžineriniu požiūriu kompiuterinės vizijos tikslas yra sukurti autonomines sistemas, kurios galėtų atlikti užduotis, kurias atlieka žmogus, o daugeliu atveju tai daryti greičiau ir efektyviau.

Kelyje matomo vaizdo segmentavimas yra svarbi problema, norint taikyti kompiuterinės regos sistemas autonominiuose automobiliuose ar savavaldžiuose robotuose. Vaizdo segmentavimo sistemos paskirtis yra kuo tiksliau aptikti ir identifikuoti kelyje esančius objektus, tokius kaip automobiliai, pėstieji, kelio ženklai ir kitas aplinkos detales. Autonominėse transporto priemonėse kompiuterinės regos sistemų kokybė ir patikimumas yra kelyje esančių asmenų saugumo klausimas. Autonominio vairavimo sistemoms, tikslus supratimas apie kitus eismo dalyvius ar kliūtis yra būtinas, norint išvengti galimų eismo įvykių. Kelyje susiduriama su daugeliu kitų eismo dalyvių, tikslus objektų ir aplinkos aptikimas yra esminis reikalavimas norint realizuoti saugias ir efektyvias autonominio vairavimo sistemas.

Tikslas ir uždaviniai

Tikslas – sukurti kelyje matomo vaizdo segmentavimo sistemą.

Uždaviniai:
1. susipažinti su egzistuojančiais vaizdo segmentavimo sprendimais;
2. išanalizuoti naudojamas technologijas, architektūras ir duomenų rinkinius;
3. sukurti ir apmokyti kompiuterinės regos vaizdo segmentavimo modelį;
4. ištestuoti vaizdo segmentavimo sistemos tikslumą;
5. išanalizuoti ir palyginti sistemos tikslumą su rinkoje esamais sprendimais.

Dokumento struktūra

Analitinėje darbo dalyje yra pateikiama informacija apie kompiuterinės regos srities apžvalgą, naudojamus vaizdo segmentavimo algoritmus, kompiuterinės regos modelių architektūras, duomenų rinkinius ir praktinį sistemų panaudojimą rinkoje. Projektinėje darbo dalyje yra pateikiama esminė informacija apie kompiuterinės regos sistemos architektūra, kuria remiantis yra realizuota programų sistema. Eksperimentinėje dalyje yra aprašomos atliktos kompiuterinės regos modelio modifikacijos, eksperimentinės sąlygos ir gauti rezultatai.

1. Analitinė dalis

Analitinės dalies skyriuje yra pateikiama informacija apie vaizdo segmentavimo problematiką, taikomus segmentavimo metodus, naudojamas architektūras, duomenų rinkinius ir praktinį kompiuterinės regos sistemų pritaikymą rinkoje.

1.1. Problematika

Svarbiausios kompiuterinės regos problemos, kurias bandoma kuo efektyviau išspręsti, yra vaizdų klasifikavimas, objektų aptikimas ir vaizdo segmentavimas. Kompiuterinės regos sritis per trumpą laiką pastebimai pagerino objektų aptikimo ir vaizdo segmentavimo rezultatus. Didelė dalis šių pasiekimų buvo pasiekta naudojant neuroninių tinklų architektūras. Skaitmeninių statinių vaizdų bei vaizdo įrašų segmentavimas yra viena pagrindinių kompiuterinės regos problemų. Žvelgiant į šią sritį, vaizdo segmentavimas yra priskiriamas prie aukšto sudėtingumo užduočių, kurios leidžia išgauti svarbią informaciją iš skaitmeninių vaizdų. Tikslus skaitmeninio vaizdo supratimas kompiuterinės regos srityje yra labai svarbus, kadangi vis daugiau taikomųjų programų remiasi šia informacija vykdant tolimesnius sprendimus [1].

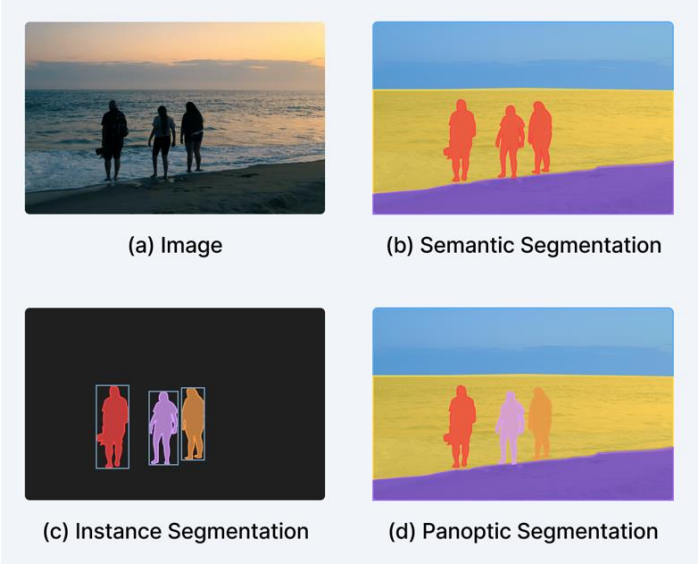
Kompiuterinės regos problemos kurį laiką buvo sprendžiamos naudojant skirtingus tradicinius kompiuterinės regos ir mašininio mokymosi metodus. Nepaisant ankstesnių metodų populiarumo, giliojo mokymosi revoliucija labai pakeitė kompiuterinės regos problemų sprendimą. Dauguma kompiuterinės regos problemų, tarp jų ir vaizdo segmentavimo yra sprendžiamos naudojant giliąsias architektūras, neuroninius tinklus. Naujos architektūros gerokai pranoksta tradicinius mašininio mokymosi sprendimus tikslumo ir efektyvumo atžvilgiu. Norint taikyti kompiuterinės regos modelius realaus laiko sistemoms, tokioms kaip autonominiai automobiliai ar savavaldžiai robotai, vaizdo segmentavimo procesas privalo būti efektyvus ir greitas. Kompiuterinės regos sistemoms tenka spręsti objektų aptikimo, krypties sekimo, klasifikavimo, segmentavimo užduotis. Vaizdo segmentavimas yra sudėtingas procesas kurį apsunkina objektų judėjimas, didelė išvaizdos įvairovė ar skirtingos formos. Objektų aptikimą tankiai apgyvendintose miesto zonose apsunkina tokie veiksniai kaip objektų panašumas, šešėliai, atspindžiai. Vaizdo segmentavimo procese pagrindinis dėmesys skiriamas skaitmeninio vaizdo padalijimui į skirtingus segmentus pagal vaizdo savybes. Atliekant vaizdo segmentavimo užduotis yra naudojami skirtingi algoritmai, kurie išskiria ir sugrupuoja pikselius. Kiekvienas pikselis yra identifikuojamas ir panašias savybes turintys vaizdo pikseliai priskiriami konkrečiam segmentui. Vėliau šie segmentai yra apibrėžiami ribomis, taip išskiriant skaitmeniniame vaizde esančius objektus nuo kitų objektų ar fono. Tai yra viena iš technikų, leidžianti išgauti prasmingą informaciją iš skaitmeninių vaizdų. Pagrindinis vaizdo segmentavimo tikslas – išanalizuoti skaitmeninį vaizdą ir identifikuoti objektus bei aplinkos detales. Atlikus vaizdo segmentavimą galima suprasti, kad tam tikri objektai priklauso atskiriems segmentams. Atliekant vaizdo segmentavimą, kai kontrasto skirtumas tarp fono ir skirtingų objektų yra didelis, kompiuterinės regos modeliams yra lengviau išskirti atskirus skaitmeninio vaizdo segmentus. Tačiau problema atsiranda tada, kai skaitmeniniuose vaizduose fonas yra chaotiškas ir skirtumai tarp objektų ir fono nėra dideli. Šiuo atveju yra žymiai sunkiau pasiekti gerą segmentavimo sistemos tikslumą. Šiuo metu vaizdo segmentavimas kompiuterinės regos srityje kelia daug iššūkių, norint tiksliai nustatyti skaitmeniniame vaizde esančius skirtingus objektus. Naujausi tyrimai rodo, kad gerai atlikta skaitmeninio vaizdo klasifikacija, lemia geresnius vaizdo segmentavimo rezultatus [2].

1.2. Srities apžvalga

Kompiuterinė rega yra kompiuterių mokslo sritis, kurioje pagrindinis dėmesys skiriamas objektų atpažinimui ir apdorojimui skaitmeniniuose vaizduose. Vaizdo segmentavimas yra skaitmeninio vaizdo apdorojimo procesas, kurio metu skaitmeninio vaizdo dalys yra suskirstomos į atskirus segmentus. Vaizdo segmentavimo metu kiekvienas skaitmeniniame vaizde esantis pikselis yra identifikuojamas ir panašias vaizdo savybes turintys pikseliai sugrupuojami į tas pačias klases. Kompiuterinės regos užduotims atlikti yra naudojami giliojo mokymosi modeliai, neuroniniai tinklai bei duomenų rinkiniai šiems modeliams apmokyti. Giliojo mokymosi modeliams duomenų rinkinys ir neuroninis tinklas yra dvi svarbios dalys. Neuroninio tinklo bei duomenų rinkinio pasirinkimas ir jo kokybė lemia sistemos tikslumą atliekant kompiuterinės regos užduotis [3]. Nemažai mokslinių tyrimų sričių pasiekė puikių rezultatų nuolat tobulinant neuroninius tinklus. Giliojo mokymosi modeliai naudoja neuroninius tinklus objektams aptikti, klasifikuoti, segmentuoti. Dėl dirbtinio intelekto pažangos ir giliojo mokymosi bei neuroninių tinklų naujovių ši sritis pastaraisiais metais sugebėjo pralenkti žmones užduotyse, susijusiose su objektų aptikimu, klasifikavimu, segmentavimu. Vienas iš veiksnių, skatinančių kompiuterinės regos sistemų vystymąsi, yra didelis generuojamų duomenų kiekis, kuris naudojamas kompiuterinės regos modeliams apmokyti ir juos tobulinti.

1.2.1. Vaizdo segmentavimo metodai

Vaizdo segmentavimo metodus galima suskirstyti į tris skirtingas grupes: semantinio segmentavimo (angl. *semantic segmentation*), objektinio segmentavimo (angl. *instance segmentation*) ir panoptinio segmentavimo (angl. *panoptic segmentation*) (žr. 1 pav.).



1 pav. Vaizdo segmentavimo tipai

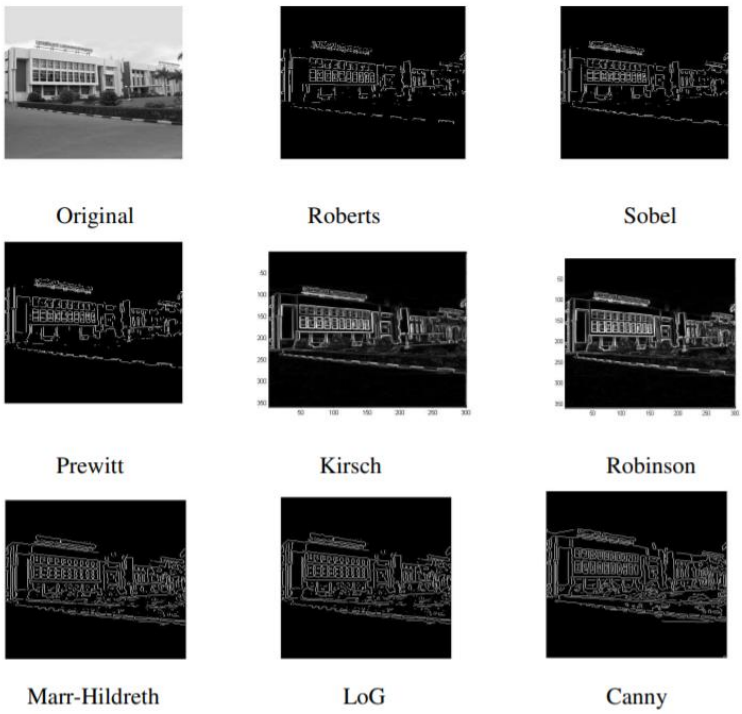
Semantinio segmentavimo algoritmai identifikuoja skaitmeniniame vaizde esančius objektus ir aplinkos detales. Objektinio segmentavimo algoritmai identifikuoja tik tam tikrus skaitmeniniame vaizde esančius objektus be aplinkos detalių. Panoptinio segmentavimo algoritmai yra patys informatyviausi, kurie sujungia objektinį ir semantinį segmentavimą. Šio tipo algoritmai identifikuoja skaitmeniniame vaizde esančius objektus ir aplinkos detales bei papildomai išskiria kiekvieną tos pačios klasės objektą. Semantinio segmentavimo atveju skaitmeniniame vaizde esantys pikseliai yra

klasifikuojami į skirtingas klases. Tam tikrai klasei priklausantys vaizdo pikseliai priskiriami konkrečiai klasei, neišskiriant tos pačios klasės skirtingų objektų. Gatvėje esančios pėsčiųjų minios vaizdas semantinio segmentavimo atveju priskiriamas vienai pėsčiųjų klasei, neišskiriant atskirų žmonių kaip objektų. Objektinio segmentavimo atveju vaizdo pikseliai klasifikuojami į skirtingas kategorijas pagal pavyzdžius, o ne pagal klases. Šis segmentavimo algoritmas neturi supratimo apie klasę kuriai priklauso klasifikuojamas regionas, tačiau gali atskirti persidengiančius ar labai panašius objektų regionus pagal jų ribas. Naudojant šį algoritmą gatvėje esančiai pėsčiųjų miniai segmentuoti, jis gali atskirti kiekvieną asmenį kaip atskirą objektą. Panoptinis segmentavimas yra plačiai naudojamas segmentavimo metodas, kuris išskirią kiekvieną vaizde esantį objektą ir aplinkos detales. Šio segmentavimo atveju vaizdas yra skirstomas pagal kategorijas, atskirus objektus ir aplinkos detales. Panoptinio segmentavimo algoritmai yra naudojami autonominiuose automobiliuose, kai vaizdo sraute reikia užfiksuoti daug informacijos apie aplinką.

Kompiuterinėje regoje neuroniniai tinklai yra viena iš svarbių sudedamųjų dalių, atliekant vaizdo segmentavimo užduotis. Pastarąjį dešimtmetį kompiuterinės regos užduotims atlikti buvo naudojamos neuroninių tinklų architektūros, tokios kaip *DenseNet, VGGNet, SegNet, AlexNet* ar *ResNet*. Dauguma šių architektūrų turi skirtingą sluoksnių skaičių. Architektūros, turinčios mažesnį sluoksnių skaičių, gali mokytis greičiau. Didesnį sluoksnių skaičių turinčios architektūros mokosi lėčiau, tačiau jos gali pasiekti geresnį tikslumą, atliekant kompiuterinės regos užduotis [4]. Šiems neuroniniams tinklams apmokyti yra reikalingi kokybiški duomenų rinkiniai. Apmokant kompiuterinės regos modelius yra naudojami tokie duomenų rinkiniai kaip *Cityscapes, Pascal Voc, Gta 5, Ade20k, Coco, CamVid,* ar *Mnist*. Modeliai, kurie atlieka vaizdo segmentavimo užduotis, dažniausiai yra vertinami pagal persidengimo (angl. *mean intersection over union*) ir pikselių tikslumo (angl. *pixel accuracy*) metrikas, kurios leidžia nustatyti esamo modelio tikslumą.

Vaizdo segmentavimo užduotims atlikti ilgą laiką buvo naudojami tradiciniai mašininio mokymosi metodai: slenkstinis, briaunų aptikimo metodas, klasterizavimu grindžiamas metodas bei dauguma kitų. Slenkstinis metodas tai vienas iš paprasčiausių lygiagretaus segmentavimo metodų. Šis segmentavimo algoritmas išskirsto skaitmeniniame vaizde esančią informaciją pagal nustatytą pilkos spalvos slenkstį. Metodas yra pranašus tuo, kad jis yra paprastas ir greitas. Skaitmeniniame vaizde esant tarp objekto ir fono dideliam kontrastui, taikant šį metodą galima pasiekti puikų segmentavimo efektą. Vienas iš šio metodo trūkumų yra tai, kad taikant jį nekontrastingiems vaizdams galimi dideli segmentavimo netikslumai [5].

Briaunų aptikimo metodas buvo vienas iš dažniausiai naudojamų algoritmų vaizdo segmentavimo užduotims atlikti. Apdorojant vaizdą, šis metodas identifikuodavo skaitmeninio vaizdo taškus, kurie turi žymių vaizdo ryškumo pokyčių. Taškai, kuriuose skiriasi vaizdo ryškumas, yra identifikuojami kaip atskirų regionų briaunos [6]. Metodas buvo plačiai naudojamas objektų aptikimo užduotyse. Vieni iš briaunų aptikimo metodų yra *Roberts, Sobel, Prewitt, Kirsch, Robinson, Marr-Hildreth, LoG* ir *Canny* (žr. 2 pav.).

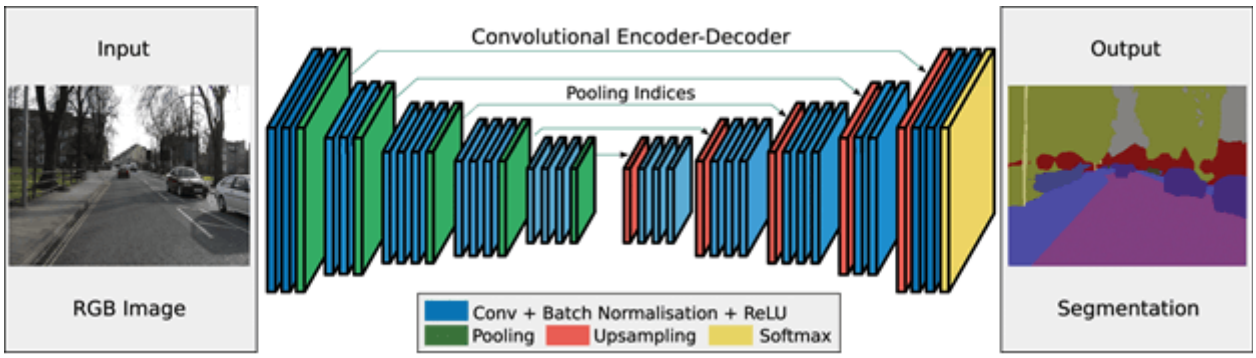


2 pav. Briaunų aptikimo metodai [6]

Klasterizavimu grindžiamame metode vienas iš dažnai naudotų metodų yra k-vidurkių (angl. *K-Means)* algoritmas. Taikant k-vidurkių klasterizavimą, algoritmas sugrupuoja pateiktus duomenis į atskiras grupes atsižvelgiant į duomenų atstumą iki grupės centrų. Šis algoritmas susideda iš dviejų atskirų fazių. Pirmoje fazėje yra nustatomas grupių skaičius, antroje kiekvienas duomenų taškas yra priskiriamas prie arčiausiai esančios grupės centro. Tradicinių mašininio mokymosi metodų sėkmė sulėtėjo tada, kai buvo pradėti naudoti giliojo mokymosi modeliai kompiuterinės regos užduotims atlikti. Norint pasiekti išskirtinį našumą atliekant kompiuterinės regos užduotis, giliojo mokymosi modeliams reikia tik kokybiškų duomenų, lyginant su tradiciniais mašininio mokymosi metodais, kuriems vien tik duomenų nepakanka.

1.2.2. Kompiuterinės regos modelių architektūros

Tarp skirtingų giliojo mokymosi modelių konvoliuciniai neuroniniai tinklai pasiekė puikų efektyvumą atliekant tokias skirtingas kompiuterinės regos užduotis kaip vaizdo klasifikavimas, objektų aptikimas ar skaitmeninio vaizdo segmentavimas. Konvoliuciniai neuroniniai tinklai per pastarąjį dešimtmetį tapo vieni iš sėkmingiausių ir plačiausiai naudojamų giliojo mokymosi architektūrų, atliekant kompiuterinės regos užduotis. Konvoliucinių neuroninių tinklų architektūros vaizdo segmentavimo užduotims atlikti naudoja kodavimo ir dekodavimo modulius (žr. 3 pav.). Kodavimo įrenginiai naudojami įvesties informacijai užkoduoti. Ši informacija yra siunčiama per tinklą, o vėliau dekoderiai naudojami vaizdui iššifruoti atgal. Kodavimo įrenginiai gali būti konvoliuciniai neuroniniai tinklai, o dekoderiai gali būti pagrįsti dekonvoliuciniais arba transponuotais neuroniniais tinklais, siekiant sukurti segmentavimo žemėlapį.

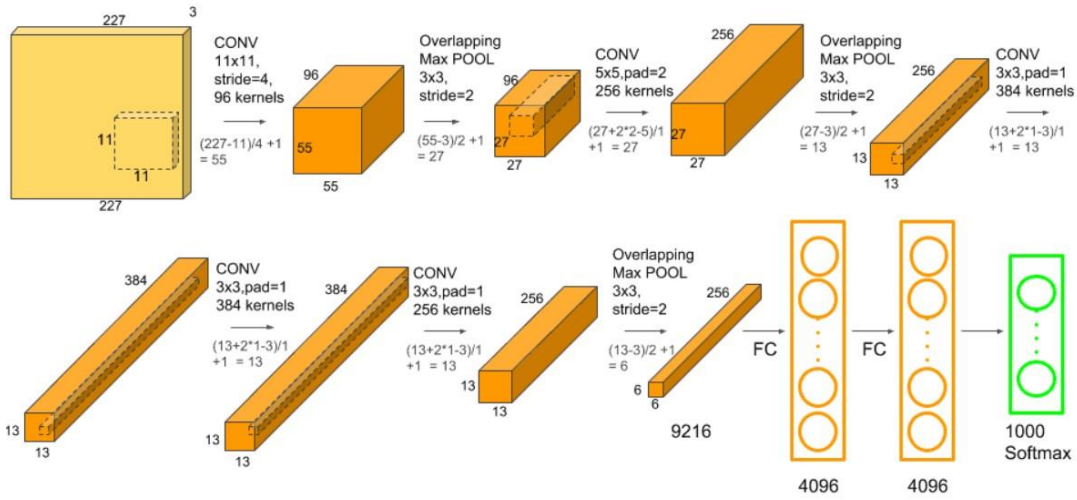


3 pav. Kodavimo ir dekodavimo architektūra [7]

1989 metais prancūzų mokslininkas Yann'as LeCun'as sukūrė vieną iš pirmųjų konvoliucinių neuroninių tinklų, kurio pavadinimas yra *LeNet-5*. Šis neuroninis tinklas buvo sukurtas ranka rašytiems skaičiams atpažinti. *LeNet-5* architektūros atsiradimas atvėrė kelią nuolatinei konvoliucinių neuroninių tinklų sėkmei, vykdant aukšto sudėtingumo kompiuterinės regos užduotis, bei paskatino mokslininkus ištyrinėti konvoliucinių neuroninių tinklų galimybes, atliekant vaizdo segmentavimo užduotis [8]. Šios architektūros dažniausiai susideda iš trijų tipų sluoksnių: konvoliucinių sluoksnių (angl. *convolutional layers*), telkimo sluoksnių (angl. *pooling layers*) ir visiškai sujungtų sluoksnių (angl. *fully-connected layers*) [9]. Konvoliuciniai sluoksniai yra vieni iš pirmųjų sluoksnių, esančių konvoliucinio tinklo architektūroje, ir yra naudojami išgauti svarbias skaitmeninio vaizdo savybes iš įvesties vaizdų. Šiame sluoksnyje matematinės konvoliucinės operacijos atliekamos tarp įvesties vaizdo ir tam tikro dydžio filtro. Atlikus šias konvoliucines operacijas yra suformuojamas savybių žemėlapis, kuris suteikia išsamesnės informacijos apie vaizdą. Vėliau šis savybių žemėlapis perduodamas kitiems sluoksniams svarbioms įvesties vaizdo savybėms išgauti. Telkiamieji sluoksniai daugumoje atveju yra naudojami iš karto po konvoliuciniais sluoksniais. Pagrindinis šių sluoksnių tikslas yra sumažinti suformuota savybių žemėlapį, tuo pačiu sumažinant skaičiavimo sąnaudas. Šis procesas atliekamas mažinant ryšius tarp sluoksnių. Egzistuoja keletas nuo naudojamo metodo priklausančių telkimo operacijų: didžiausio elemento (angl. *max pooling*), vidurkio (angl. *average pooling*) ir mažiausio elemento (angl. *min pooling*). Telkiamieji sluoksniai konvoliucinių neuroninių tinklų architektūrose yra naudojami tarp konvoliucinių sluoksnių ir visiškai sujungtų sluoksnių. Visiškai sujungti sluoksniai yra naudojami neuronams sujungti tarp dviejų skirtingų sluoksnių. Šiame etape yra pradedamas klasifikavimo procesas. Šie sluoksniai dažniausiai yra prieš išvesties sluoksnį ir sudaro paskutinius kelis konvoliucinių neuroninių tinklų architektūros sluoksnius.

Kompiuterinės regos modelis *AlexNet*, paremtas konvoliucinio neuroninio tinklo architektūra, 2012 m. laimėjo didelio masto vaizdinio atpažinimo iššūkį (angl. *large scale visual recognition challenge*). Tai konkursas, kurio metu mokslininkų komandos vertina savo algoritmus, naudodamos didžiulį pažymėtų vaizdų rinkinį *ImageNet*. Dalyviai taip pat rungtyniauja tarpusavyje, siekdami didesnio tikslumo atliekant įvairias kompiuterinės regos užduotis. *AlexNet* architektūrą sudaro aštuoni sluoksniai: penki – konvoliuciniai, o trys – pilnai sujungti. Pirmieji du architektūros *AlexNet* (žr. 4 pav.) konvoliuciniai sluoksniai turi persidengiančius telkiamuosius sluoksnius. Trečiasis, ketvirtasis ir penktasis konvoliuciniai sluoksniai tiesiogiai sujungti vienas su kitu. Penktasis *AlexNet*

architektūros konvoliucinis sluoksnis turi persidengiantį telkimo sluoksnį, prijungtą prie pilnai sujungtų sluoksnių.



4 pav. *AlexNet* architektūra [10]

Pilnai sujungti architektūros *AlexNet* sluoksniai turi po 4096 neuronus. Antrasis pilnai sujungtas sluoksnis patenka į *Softmax* klasifikatorių, turintį 1000 klasių. *AlexNet* yra galingas modelis, galintis pasiekti aukštą tikslumą išbandant jį su sudėtingais duomenų rinkiniais. Tačiau pašalinus bet kurį konvoliucinį sluoksnį, modelio tikslumas smarkiai pablogėja [10]. Manoma, kad šie neuroniniai tinklai pasiekė puikių rezultatų kompiuterinės regos srityje dėl savo architektūroje naudojamų konvoliucinių operacijų.

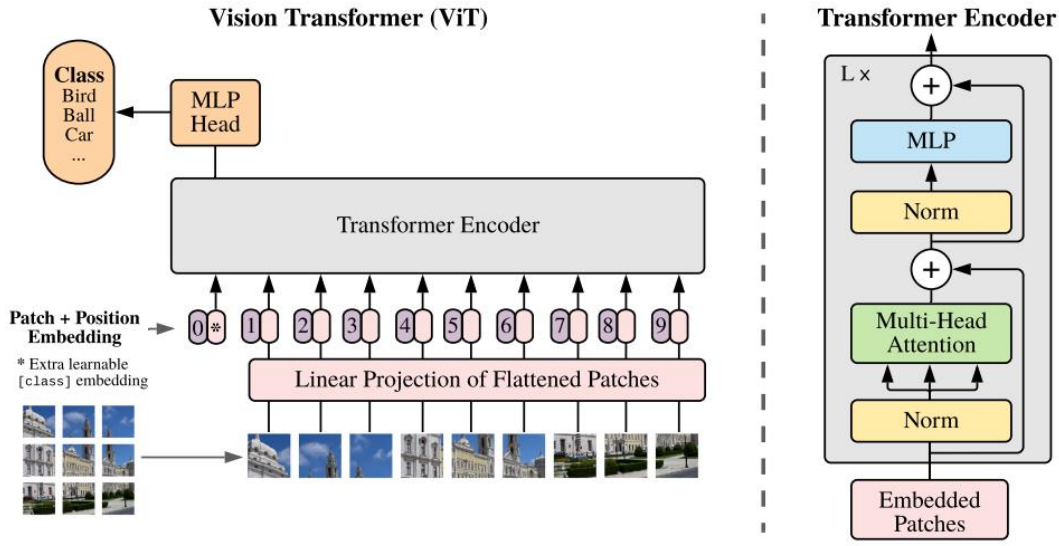
Pastaruoju metu vaizdo transformatoriai (angl. vision transformers) pasirodė kaip konkurencinga alternatyva ilgą laiką naudotiems konvoliuciniams neuroniniams tinklams, atliekant kompiuterinės regos užduotis. Transformatoriaus (angl. transformer) neuroninį tinklą sukūrė ir pristatė mokslininkas Ashish‘as Vaswani 2017 metais [11]. Šie neuroniniai tinklai šiandien savo efektyvumu ir tikslumu varžosi su moderniausiomis konvoliucinių neuroninių tinklų architektūromis. Naujausi tyrimai rodo, kad sukurti sėkmingus kompiuterinės regos modelius galima ir nenaudojant konvoliucinių sluoksnių. Viena iš tokių idėjų yra naudoti vaizdo transformatorių neuroninius tinklus, kurie taiko dėmesiu pagrįstą architektūrą įvesties vaizdams ir pasiekia konkurencingą efektyvumą, atliekant įvairias kompiuterinės regos užduotis [12]. Vaizdo transformatorių architektūroje skaitmeninis vaizdas pirmiausia suskaidomas į mažesnes dalis (žr. 5 pav.).

5 pav. Vaizdo suskirstymas į dalis [13]

Sekančiame etape šios vaizdo dalys paverčiamos į vientisą vaizdų seką, kaip yra pavaizduota šeštame paveikslėlyje (žr. 6 pav.). Be to, kiekviena vaizdo dalis sekos sudėtyje turi pozicinį žymėjimą.

6 pav. Vaizdų seka [13]

Šis neuroninis tinklas neturi supratimo, kurioje vietoje turi būti kiekviena dalis, todėl pozicinis vaizdo dalių žymėjimas padeda nustatyti skirtingų vaizdo dalių vietas. Be to, pozicinio žymėjimo skaičiai neuroniniam tinklui yra kaip mokomieji vektoriai [13]. Šie vektoriai kartu su vaizdo dalimis perduodami į neuroninį tinklą (žr. 7 pav.).



7 pav. Vaizdo transformatoriaus architektūra [13]

Remiantis moderniausių architektūrų rezultatais, atliekant vaizdo segmentavimo užduotis su *ADE20K* duomenų rinkiniu, žinoma, kad pirmąją vietą užima *FD-SwinV2-G* transformatoriaus neuroninio tinklo modelis. Šį modelį sudaro trys milijardai parametrų. Atliekant semantinio segmentavimo užduotį su *ADE20K* duomenų rinkiniu, jis pasiekia 61,4 % tikslumo rezultatą, nustatydamas naują rekordą šioje užduotyje [14].

1 lentelė. Modelių rezultatai su *ADE20K* duomenų rinkiniu

Modelio pavadinimas	**Parametrai (mln.)**	**mIoU (%)**
FD-SwinV2-G	3000	61,4
Mask DINO	223	60,8
ViT-Adapter-L	571	60,5
SwinV2-G	3000	59,9
ViT-Adapter-L	451	58,4

Pirmoje lentelėje pateikti tiksliausi kompiuterinės regos modeliai, pasižymintys skirtingais parametrų dydžiais ir pasiektais *mIoU* metrikos tikslumo rezultatais. Šie modeliai pasiekia aukščiausius semantinio segmentavimo rezultatus, naudojant *ADE20K* duomenų rinkinį. Visi modeliai paremti transformatoriaus neuroninio tinklo architektūra. Antroje lentelėje pateikti kompiuterinės regos modeliai su jų efektyvumo metrikomis, atliekant semantinio segmentavimo užduotis, naudojant *Cityscapes* duomenų rinkinį.

2 lentelė. Modelių rezultatai su *Cityscapes* duomenų rinkiniu

Modelio pavadinimas	**mIoU (%)**
HRNetV2-OCR+PSA	86,9
HRNet-OCR	86,3
ViT-Adapter-L	85,8
SeMask	85,0
VOLO-D4	84,3

Remiantis antroje lentelėje pateiktais penkių tiksliausių kompiuterinės regos modelių duomenimis, matyti, kad pirmąją ir antrąją vietas užima *HRNetV2-OCR+PSA* ir *HRNet-OCR* modeliai, pasiekę atitinkamai 86,9 % ir 86,3 % tikslumo rezultatus. Šie modeliai sukurti remiantis konvoliucinio neuroninio tinklo architektūros principu. Likusieji lentelėje pateikti modeliai sukurti remiantis transformatoriaus neuroninio tinklo architektūros principu [14, 15].

1.2.3. Duomenų rinkiniai

Norint sukurti tikslius kompiuterinės regos giliojo mokymosi modelius, svarbu turėti kokybiškus duomenų rinkinius, kurie lemia modelių efektyvumą. Kokybiškų duomenų trūkumas yra viena iš pagrindinių kliūčių, stabdančių spartesnę giliojo mokymosi pažangą. Dauguma mašininio mokymosi algoritmų mokosi iš daugybės jiems pateikiamų duomenų. Efektyviai veikiantys kompiuterinės regos modeliai veikia dėl didelio kiekio savarankiškai sužymėtų skaitmeninių vaizdų, naudojamų jų apmokymo metu. Dabartinė giliojo mokymosi era pradėjo formuotis po reikšmingo įvykio, kai buvo pristatytas *ImageNet* duomenų rinkinys. Šis duomenų rinkinys buvo naudojamas 2012 metais didelio mąsto vizualinio atpažinimo iššūkyje. *ImageNet* rinkinį sudaro daugiau nei 14 milijonų sužymėtų didelės raiškos vaizdų, priklausančių beveik 22 tūkstančiams kategorijų [16]. Šis duomenų rinkinys naudojamas testuojant kompiuterinės regos modelių tikslumą atliekant vaizdo klasifikavimo ir objektų aptikimo užduotis.

Norint sukurti efektyvią kelyje matomo vaizdo segmentavimo sistemą, modelio apmokymui reikalingi specifiniai šios srities duomenų rinkiniai. Atliekant kompiuterinės regos modelio apmokymą, duomenų rinkinyje turi būti tokie objektai ir aplinkos detalės kaip transporto priemonės, pėstieji, šaligatviai, kelio ženklai, šviesoforai, pastatai ir daugelis kitų objektų, matomų kelyje. Efektyvus kelyje matomo vaizdo segmentavimas ir objektų aptikimas yra viena iš svarbiausių užduočių, realizuojant autonomines vairavimo sistemas. Dėl vis didėjančių duomenų kiekių svarbu atsirinkti kokybiškus duomenų rinkinius, siekiant sukurti efektyvią kelyje matomo vaizdo segmentavimo sistemą. Trečioje lentelėje pateikti populiariausi anotuoti duomenų rinkiniai, skirti kompiuterinės regos modelių apmokymui.

3 lentelė. Duomenų rinkiniai

Pavadinimas	**Vaizdų skaičius**
Cityscapes	25 000
Mapillary Vistas	25 000
GTA 5	24 966
ADE20K	20 000

Atliekant kompiuterinės regos kelyje matomo vaizdo semantinio segmentavimo užduotis, *Cityscapes* yra vienas iš populiariausių duomenų rinkinių. Šis rinkinys susideda iš 5000 aukštos kokybės tiksliai sužymėtų vaizdų ir papildomų 20 000 grubias anotacijas turinčių skaitmeninių vaizdų. Vaizdai, esantys Cityscapes duomenų rinkinyje, buvo įrašyti 50 skirtingų Vokietijos miestų [17]. Šis duomenų rinkinys yra vienas iš dažniausiai naudojamų apmokant kompiuterinės regos modelius kelyje matomo vaizdo segmentavimo užduotims atlikti. *Cityscapes* rinkinį sudaro 19 skirtingų kategorijų, tokių kaip šaligatviai, keliai, automobiliai, sunkvežimiai, motociklai ar pastatai. Pastaraisiais metais atliekant semantinio segmentavimo modelių tikslumo bandymus, naudojant *Cityscapes* duomenų rinkinį, buvo pasiektas geriausias 85,2 % tikslumo rezultatas.

Mapillary Vistas yra dar vienas didesnės apimties vaizdo segmentavimo duomenų rinkinys, kurį sudaro apie 25 000 aukštos kokybės skaitmeninių vaizdų, anotuotų į 66 objektų kategorijas. Šis duomenų rinkinys yra 5 kartus didesnis, lyginant su *Cityscapes* duomenų rinkiniu, pagal sužymėtus aukštos kokybės vaizdus. Skaitmeniniai vaizdai duomenų rinkinyje buvo užfiksuoti nepriklausomai nuo oro sąlygų, paros meto ar sezono. Vaizdams užfiksuoti buvo naudojami mobilieji telefonai, planšetiniai kompiuteriai ar veiksmo kameros. Kuriant šį duomenų rinkinį buvo atsižvelgta į objektų įvairovę, detalių gausą ir geografinį mastą. *Mapillary Vistas* duomenų rinkinys sukurtas atsižvelgiant į išaugusį susidomėjimą autonominėmis transporto priemonėmis ir siekiant sėkmingai plėtoti efektyviausius kompiuterinės regos modelius, atliekant vaizdo segmentavimo užduotis [18].

GTA 5 yra semantinio segmentavimo duomenų rinkinys, turintis 24 966 anotuotus sintetinius vaizdus. Šis duomenų rinkinys yra išskirtinis tuo, kad vaizdai, esantys rinkinyje, buvo išgauti iš kompiuterinio žaidimo *Grand Theft Auto*. Šis rinkinys turi 19 semantinių klasių. Vaizdai esantys rinkinyje, yra užfiksuoti iš vairuotojo vaizdo perspektyvos [19]. Apžvelgiant pastarųjų kelių metų semantinio segmentavimo modelių rezultatus, naudojant šį duomenų rinkinį, buvo rasti aštuoni modeliai, iš kurių geriausias pasiekia 73,8 % vidutinės persidengimo metrikos tikslumo rezultatą.

ADE20K yra dar vienas populiarus semantinio segmentavimo duomenų rinkinys, susidedantis iš daugiau kaip 20 000 anotuotų vaizdų. Duomenų rinkinyje yra 150 semantinių kategorijų, tokių kaip dangus, kelias, žolė, automobiliai ar žmonės. Įdomus faktas yra tai, kad bet kuriame *ADE20K* duomenų rinkinio vaizde yra mažiausiai 5 objektai, o didžiausias objektų skaičius viename vaizde siekia 273 [20]. Šis duomenų rinkinys buvo anotuotas vieno eksperto, pateikiant itin išsamias vaizdo anotacijas. Remiantis pastarųjų metų semantinio segmentavimo modelių bandymais, galima teigti, kad šis rinkinys dažnai naudojamas tikrinant moderniausių kompiuterinės regos modelių efektyvumą. Testuojant kompiuterinės regos modelius su šiuo rinkiniu, buvo pasiektas geriausias 61,4 % vidutinės persidengimo metrikos tikslumo rezultatas.

1.3. Taikymų apžvalga

Kompiuterinės regos vaizdo segmentavimo sprendimai yra plačiai naudojami robotikoje, medicinoje ir automobilių pramonėje, atliekant skaitmeninio vaizdo apdorojimo užduotis. Vaizdo segmentavimas medicinoje sėkmingai taikomas tiek diagnostikos, tiek gydymo sektoriuose. Pritaikius kompiuterinės regos sprendimus kompiuterinės tomografijos ar magnetinio rezonanso tyrimuose, galima efektyviai identifikuoti įvairias ligas iš skaitmeninių medicininių vaizdų. Nepaisant iššūkių dėl žemo kontrasto skaitmeniniuose vaizduose, segmentavimo sprendimai daugeliu atvejų taikomi kompiuterinės tomografijos organų skenavimui, rentgeno nuotraukoms ir skaitmeninės patologijos ląstelių segmentacijai.

Vaizdo segmentavimo sprendimai taip pat naudojami daugelyje robotikos sričių – nuo pramonės iki žemės ūkio ir paslaugų sektoriaus. Kiekviena robotikos šaka vis labiau priklauso nuo kompiuterinės regos sprendimų. Robotams reikia aiškaus pikselių lygio supratimo apie juos supančią aplinką, kad jie galėtų efektyviai atlikti tam tikrus veiksmus. Vaizdo segmentavimo sprendimai robotams padeda suprasti juos supantį pasaulį taip, kaip jį mato žmogus. Automatizavimas ir robotika yra viena iš pirmaujančių sričių, praktiškai pritaikant naujausius kompiuterinės regos sprendimus [21].

Autonominėse transporto priemonėse taikomi kompiuterinės regos, mašininio mokymosi ir kiti pažangūs sprendimai. Vaizdo segmentavimas yra kritiškai svarbus autonominių transporto priemonių komponentas. Vairuodamas automobilį, vairuotojas turi atkreipti dėmesį į kelią, šaligatvius, kelio ženklus, pėsčiuosius ir visas kitas transporto priemones. Autonominių transporto priemonių gamintojai, siekdami užtikrinti saugumą, turi pasirūpinti visomis šių sistemų stebėjimo ir analizės galimybėmis. Be to, kad galėtų realiu laiku matyti, interpretuoti ir reaguoti į aplinkos scenarijus, šios transporto priemonės turi turėti itin detalų aplinkos vaizdą. Autonominės transporto priemonės atlieka važiuojamosios kelio dalies, transporto priemonių, pėsčiųjų ir kitų aplinkos objektų segmentaciją. Efektyvūs vaizdo segmentavimo sprendimai leidžia autonominėms transporto priemonėms saugiai dalyvauti eisme. Remiantis JAV nacionalinės greitkelių eismo saugumo administracijos atliktu tyrimu nustatyta, kad 94 % visų eismo įvykių įvyksta dėl žmogiškojo faktoriaus klaidos [22].

Taikant kompiuterinės regos sistemas automobilių pramonėje, tie patys vaizdo segmentavimo algoritmai gali būti naudojami ir su kitomis sistemomis robotikos ir aviacijos srityse. Pavyzdžiui, atliekant *NASA* 2020 m. vykdytą misiją, robotas *Perseverance* naudojo kompiuterinės regos sistemą saugiam nusileidimui, aptikdamas paviršiaus reljefą ir autonomiškai pasirinkdamas saugiausią nusileidimo vietą [23]. Semantinio segmentavimo algoritmų naudojimas gali padidinti aplinkos aptikimo tikslumą [24]. Siekiant realizuoti visiškai autonomines transporto priemones ateityje, tikslios aplinkos aptikimo sistemos tapo neatskiriamais komponentais, užtikrinant tinkamą sudėtingos ir dinamiškos aplinkos stebėjimą ir interpretavimą [25]. Semantiniai duomenys gali sumažinti roboto priklausomybę nuo neapdorotų jutiklių duomenų ir išorinių GPS signalų, suteikdami papildomą informaciją apie aplinką [26].

Naviguodami sudėtingoje miesto aplinkoje, autonominiai robotai reikalauja tikslaus aplinkos suvokimo ir patikimo maršruto planavimo, turėdami ribotus skaičiavimo resursus. Autonominio roboto gebėjimas suprasti jį supančią aplinką yra pagrindas, leidžiantis autonominėms sistemoms spręsti įvairias sudėtingesnes problemas [27]. Autonominių sistemų efektyvumas labai priklauso nuo jų gebėjimo naviguoti sudėtingose ir nestruktūruotose aplinkose [28]. Naujausi kompiuterinės regos algoritmai labai pagerino robotų navigaciją, suteikiant galimybę realiuoju laiku vykdyti svarbias užduotis: aplinkos suvokimą, kliūčių aptikimą ir vengimą, maršruto planavimą bei sekimą [29]. Planuojant maršrutą, kliūčių vengimas yra esminė robotikos užduotis, nes autonominis roboto veikimas reikalauja, kad jis pasiektų tikslą be susidūrimų [30]. Objektų aptikimo strategijos turi gebėti atpažinti tiek statines kliūtis, tokias kaip infrastruktūra ar stovinčios transporto priemonės, tiek dinamines kliūtis, įskaitant pėsčiuosius ir judančias transporto priemones, nes kiekviena iš jų kelia unikalius saugios navigacijos iššūkius. Siekiant išvengti galimo susidūrimo su pėsčiaisiais, tiksli objektų aptikimo sistema leidžia autonominiams robotams laiku reaguoti ir sumažinti nelaimingų atsitikimų riziką [31]. Tiksli kompiuterinės regos sistema yra būtina siekiant užtikrinti saugų ir efektyvų autonominių robotų veikimą, ypač dinamiškoje miesto aplinkoje. Aplinkos supratimas per vaizdinę informaciją, naudojant kompiuterinio matymo metodus, mašininį mokymąsi ir įvairius algoritmus, yra pagrindinis šiuolaikinių sprendimų tikslas [32].

Autonominių pristatymo robotų veikimas viešose erdvėse, kuriose gyvena ir dirba žmonės, reikalauja gebėjimo atpažinti ir sekti saugius bei tinkamus maršrutus, siekiant užtikrinti efektyvų ir socialiai priimtiną robotų judėjimą [33, 34]. Didėjant robotų ir žmonių sąveikai kasdienėje aplinkoje, saugumo užtikrinimas tampa esminiu veiksniu [35]. Kompiuterinės regos sistemų taikymas autonominėse transporto priemonėse padeda didinti jų efektyvumą, intelektualumą ir eismo saugumą [36]. Autonominių robotų regos technologijų vystymasis yra reikšmingas žingsnis į priekį, leidžiantis geriau suprasti autonominių sistemų galimybes ir jų poveikį žmonių gyvenimui [37].

1.3.1. Perspektyviausi taikymai

Vaizdo segmentavimas yra vienas iš svarbiausių skaitmeninio vaizdo apdorojimo procesų ir pastaraisiais metais plačiai naudojamas automobilių pramonėje bei robotikoje. Autonominis vairavimas artimiausioje ateityje bus viena iš revoliucinių technologijų, kuri turės didelę įtaką kasdieniam žmonių gyvenimui. Vaizdo segmentavimo sistemos autonominiams automobiliams suteikia aplinkinį pasaulio matymą ir yra kritiškai svarbios realizuojant saugų autonominį transporto priemonių vairavimą. Viena iš didžiausių mirties priežasčių yra automobilių avarijos. Dauguma šių nelaimingų atsitikimų įvyksta dėl žmogiškojo faktoriaus klaidos. Kai kurios iš pagrindinių mirčių keliuose priežasčių yra išsiblaškymas, greičio viršijimas ir neapdairumas. Realizuojant autonominio vairavimo sistemas yra siekiama išspręsti šią didžiulę eismo įvykių problemą. Kadangi autonominės sistemos yra užprogramuotos taip, kad važiuotu efektyviai ir saugiai, jos sumažina, o kartais ir pašalina žmogaus vairavimo poreikį, taip eliminuodamos minėtas žmogiškąsias klaidas [38]. Automobilių pramonė yra labai perspektyvi sritis, įgyvendinant kompiuterinės regos vaizdo segmentavimo sprendimus autonominėse transporto priemonėse. Kuo semantinio segmentavimo procesas yra tikslesnis ir atliekamas per trumpesnį laiką, tuo autonominės transporto priemonės tiksliau supranta jas supančią aplinką ir gali priimti saugesnius sprendimus. Suteikiant autonominėms transporto priemonėms aplinkinio pasaulio matymą taikant kompiuterinės regos sprendimus, galima gauti daug privalumų, tokių kaip didesnis saugumas kelyje, mažesnės išlaidos, patogesnės kelionės, didesnis mobilumas ir mažesnė aplinkos tarša [39].

Šiuolaikinės pramoninės automatikos ir robotikos taikymas vis labiau priklauso nuo kompiuterinės regos vaizdo segmentavimo sprendimų. Norint, kad robotai efektyviai ir saugiai veiktų nežinomoje aplinkoje, būtinas tikslus aplinkos objektų supratimas. Efektyvūs kompiuterinės regos modeliai padeda robotams sąveikauti su įvairiais, anksčiau nematytais objektais ir atlikti specifines užduotis [40]. Per ateinantį dešimtmetį dauguma pramoninių veiklų, reikalaujančių nuolatinio žmogaus įsikišimo, gali būti iš dalies ar visiškai automatizuotos. Išaugęs pramonės sektoriaus susidomėjimas kompiuterinės regos sprendimais turi ir daug privalumų. Pritaikius efektyvius kompiuterinės regos sprendimus pramoniniuose robotuose, galima sumažinti gamybos sąnaudas, išlaikyti pastovią gamybos kokybę ir padidinti produktyvumą. Be to, šie sprendimai suteikia didesnį gamybos lankstumą ir leidžia greičiau reaguoti į darbuotojų trūkumą. Efektyviai veikiantys kompiuterinės regos modeliai skatina pasitikėjimą ir tolesnes investicijas į robotika bei kompiuterinės regos sprendimus pramonės sektoriuje.

1.4. Įgyvendinimo problemos

Norint sėkmingai įgyvendinti kompiuterinės regos sprendimus, būtina naudoti kokybiškus duomenų rinkinius, skirtus kompiuterinės regos modelių apmokymui. Šiandien vis dar susiduriama su kokybiškų duomenų trūkumu. Duomenų trūkumas yra viena iš problemų, įgyvendinant efektyvius

kompiuterinės regos sprendimus. Sužymėti ir anotuoti duomenų rinkiniai yra labai svarbūs norint pasiekti didelį kompiuterinės regos modelių efektyvumą. Įgyvendinti standartinius kompiuterinės regos sprendimus, kuriems apmokyti pakanka bendros paskirties duomenų rinkinių, nėra labai sudėtinga. Tačiau susiduriama su akivaizdžiai didesniais iššūkiais kai norima pritaikyti kompiuterinės regos sprendimus tokiose srityse kaip medicina. Dažnai kyla sunkumų gaunant aukštos kokybės skaitmeninius vaizdus, tokius kaip kompiuterinės tomografijos ar rentgeno nuotraukos, dėl privatumo apribojimų. Taip pat ne visais atvejais gaunami duomenys yra kokybiški, todėl prastos kokybės ar nesužymėti duomenų rinkiniai reikalauja tolimesnių duomenų anotacijų. Todėl šiandien vis dar susiduriama su nepakankamu kiekiu kokybiškų duomenų, norint apmokyti kompiuterinės regos modelius specifinėms užduotims atlikti.

Daug laiko reikalaujantys modelių treniravimo procesai, modelių persimokymo (angl. *overfitting*) ar nykstančių gradientų (angl. *vanishing gradients*) problemos yra taip pat vienos iš iššūkių, kuriant efektyvius giliojo mokymosi modelius. Didelis tikslumas ir trumpas modelių treniravimo laikas yra pagrindiniai tikslai, siekiant sukurti efektyvius vaizdo segmentavimo sprendimus. Tačiau giliojo mokymosi architektūroms modelių treniravimas dažnai yra daug laiko ir resursų reikalaujantis procesas. Giliojo mokymosi architektūrų kūrėjai kaip vieną iš sprendimų naudoja telkimo sluoksnių (angl. *pooling layers*) integraciją, taip siekiant sumažinti modelių treniravimosi laiką. Modelių persimokymas yra dar viena problema, kuri atsiranda kai modelis per daug prisitaiko prie treniravimui skirto duomenų rinkinio. Šiuo atveju modelis tampa neefektyvus, atliekant modelio bandymą su testavimo duomenų rinkiniu. Modelių persimokymo problemos gali būti sprendžiamos padidinant mokymosi duomenų rinkinio apimtį, naudojant duomenų augmentaciją ar sumažinant esamo modelio kompleksiškumą [41].

Nykstantys gradientai yra dar viena iš pagrindinių problemų giliose architektūrose. Nors gilios architektūros pasižymi puikiu našumu, neišsprendus šios problemos sunku pasiekti efektyvių rezultatų. Sigmoidinė funkcija yra viena iš populiariausių aktyvinimo funkcijų, naudojamų giliose neuroninių tinklų architektūrose. Šios funkcijos naudojimas apriboja giliųjų neuroninių tinklų mokymąsi, nes sukelia nykstančių gradientų problemą. Dėl šios problemos neuroninis tinklas mokosi lėčiau arba kai kuriais atvejais nesimoko visiškai. Vienas iš problemos sprendimų yra pakeisti sigmoidinę tinklo aktyvinimo funkciją į *ReLU* (angl. *rectified linear unit*). Šios funkcijos naudojimas padeda išvengti nykstančių gradientų problemos, užtikrinant efektyvesnį neuroninio tinklo mokymąsi [41].

2. Projektinė dalis

Programinės įrangos architektūros specifikacijos skyriaus paskirtis yra sudaryti bendrą supratimą apie kuriamos sistemos architektūros sprendimus. Projektinėje dalyje yra pateikiami esminiai architektūriniai sprendimai, kuriais remiantis yra realizuojama programų sistema. Projektinės dalies informacija padeda suprasti programinės įrangos komponentus nesigilinant į programinį sistemos kodą. Architektūros specifikacijos sudarymui panaudotas *UML* modeliavimo įrankis *MagicDraw*, kuris leidžia atvaizduoti vidinę programų sistemos struktūrą. Skyriuje pateikiamos panaudojimo atvejų, klasių, veiklos ir sekų diagramos. Kompiuterinės regos kelyje matomo vaizdo segmentavimo sistemai sukurti naudojama transformatoriaus neuroninio tinklo architektūra.

Naudojantis *UML* modeliavimo įrankiu, programų sistemos architektūra pateikiama žemiau nurodytomis diagramomis:

- Panaudojimo atvejų diagrama ir specifikacijos;
- Sistemos išskaidymo į modulius ir klases diagrama;
- Veiklos ir sekų diagramos;
- Išdėstymo diagrama.

2.1. Funkciniai reikalavimai

Kelyje matomo vaizdo segmentavimo sistema turi suteikti naudotojams galimybę atlikti skaitmeninio vaizdo semantinį segmentavimą. Sistema turi analizuoti pateiktus vaizdus, aptikti įvairius objektus ir juos klasifikuoti į skirtingas kategorijas. Pasirinkus tinkamus vaizdo failus ir įvykdžius segmentavimo procesą, sistema privalo pateikti ir išsaugoti galutinį segmentavimo rezultatą. Kompiuterinės regos sistema privalo atlikti šiuos funkcinius reikalavimus:

- Leisti nurodyti statinį ar dinaminį vaizdo failą;
- Atlikti statinio vaizdo semantinį segmentavimą;
- Atlikti dinaminio vaizdo semantinį segmentavimą;
- Pateikti segmentavimo rezultatą ir jį išsaugoti.

2.2. Nefunkciniai reikalavimai

2.2.1. Reikalavimai sistemos išvaizdai

Kompiuterinės regos sistema bus sukurta su grafine vartotojo sąsaja, kuri bus lengvai suprantama, intuityvi ir patogi naudotis. Grafinė sąsaja suteiks vartotojui galimybę nurodyti norimą failo tipą – statinį arba dinaminį, taip pat inicijuoti vaizdo segmentavimo procesą vienu mygtuko paspaudimu. Be to, sistema leis lengvai pasiekti ir peržiūrėti išsaugotus segmentavimo rezultatus, atveriant tam skirtą failų direktoriją. Grafinės vartotojo sąsajos sukūrimas leis užtikrinti patogų ir efektyvų sistemos naudojimą.

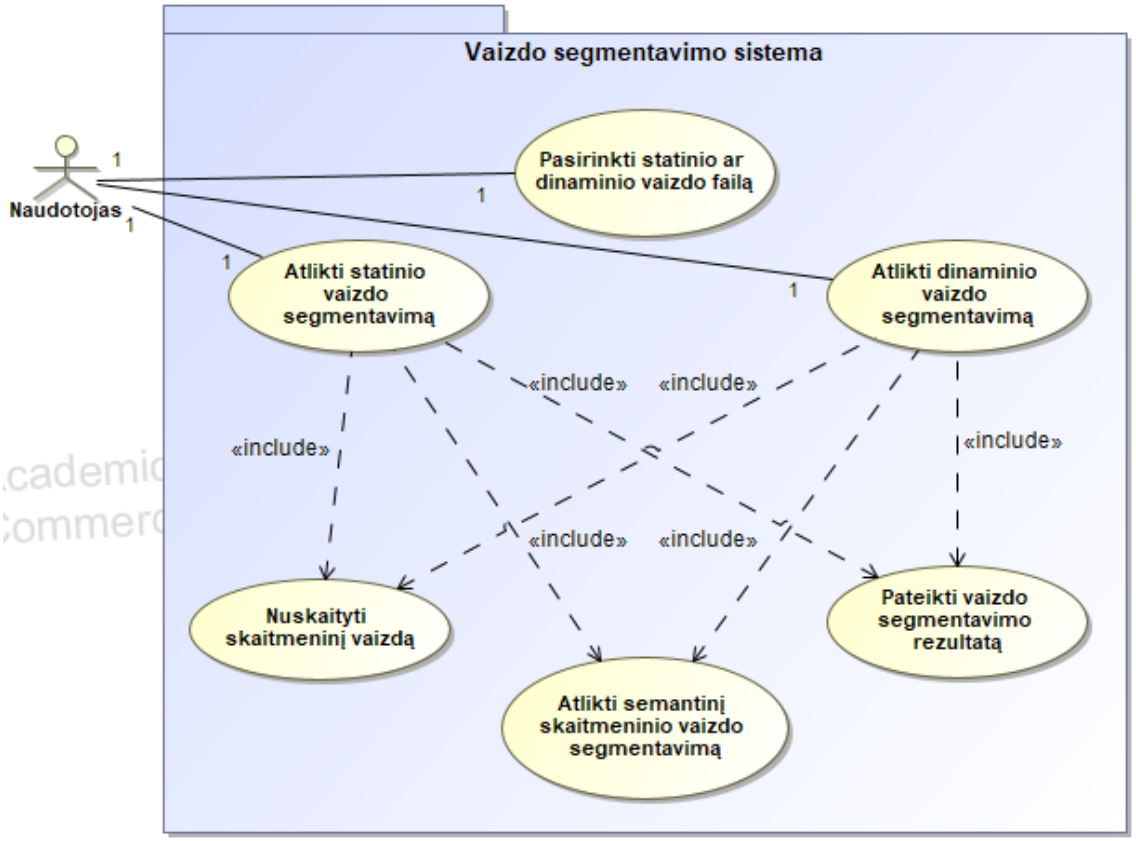
2.2.2. Reikalavimai panaudojamumui

Kompiuterinės regos vaizdo segmentavimo modelis turi būti lengvai ir aiškiai panaudojamas vartotojo, kuris yra pažengęs mašininio mokymosi ir kompiuterinės regos srityse. Sistemos panaudojimas ir supratimas, vartotojui nurodant statinio ar dinaminio vaizdo failą, turi užtrukti ne ilgiau kaip 10 minučių.

2.2.3. Reikalavimai vykdymo charakteristikoms

Kelyje matomo vaizdo segmentavimo sistemos tikslumas, atliekant testavimą su *Cityscapes* duomenų rinkiniu, turi viršyti 75 %, vertinant pagal kompiuterinės regos modelių *mIoU* tikslumo metriką.

2.3. Panaudojimo atvejai



8 pav. Panaudojimo atvejų diagrama

Šiame poskyryje pateiktose lentelėse yra išsamesnė informacija apie kiekvieną sistemos panaudojimo atvejį.

4 lentelė. Panaudojimo atvejis „Pasirinkti statinio ar dinaminio vaizdo failą“

Panaudojimo atvejis	Pasirinkti statinio ar dinaminio vaizdo failą.
Tikslas/Uždavinys	Naudotojas pasirenka skaitmeninį statinio arba dinaminio kelyje matomo vaizdo failą, kuriam bus atliekamas vaizdo segmentavimas.
Dalyviai	Naudotojas.
Prieš sąlyga	Pasirinktas failas turi būti tinkamo formato.
Sužadinimo sąlyga	Naudotojas nurodo tinkamą skaitmeninį vaizdo failą.
Po-sąlyga	Nurodžius tinkamą failą, naudotojas gali inicijuoti vaizdo segmentavimo procesą.
Pagrindinis scenarijus	Naudotojas pasirenka tinkamo formato skaitmeninį vaizdo failą. Sistema leidžia inicijuoti vaizdo segmentavimo procesą.
Alternatyvus scenarijus	Naudotojas nepasirenka arba pasirenka netinkamo formato failą. Sistema pateikia klaidos pranešimą apie nepasirinktą ar netinkamą failą.

5 lentelė. Panaudojimo atvejis „Atlikti statinio vaizdo segmentavimą“

Panaudojimo atvejis	Atlikti statinio vaizdo segmentavimą.
Tikslas/Uždavinys	Atlikti skaitmeninio statinio vaizdo segmentavimą nurodytam naudotojo failui.
Dalyviai	Naudotojas.
Prieš sąlyga	Pasirinktas tinkamo formato statinis vaizdo failas.
Sužadinimo sąlyga	Pasirinkus statinį vaizdo failą, inicijuojamas vaizdo segmentavimo procesas.
Po-sąlyga	Naudotojas gali peržiūrėti atliktą segmentavimo rezultatą pasirinktam failui.
Pagrindinis scenarijus	Naudotojas pasirenka skaitmeninį statinio vaizdo failą ir inicijuoja segmentavimo procesą.
Alternatyvus scenarijus	Naudotojas nepasirenka tinkamo statinio vaizdo failo. Sistema pateikia klaidos pranešimą apie netinkamą failą.

6 lentelė. Panaudojimo atvejis „Nuskaityti skaitmeninį vaizdą“

Panaudojimo atvejis	Nuskaityti skaitmeninį vaizdą.
Tikslas/Uždavinys	Nuskaityti skaitmeninį vaizdą iš naudotojo nurodyto failo.
Dalyviai	Naudotojas.
Prieš sąlyga	Nurodytas tinkamas skaitmeninis vaizdo failas.
Sužadinimo sąlyga	Failų sistemoje pasirenkamas tinkamas skaitmeninio vaizdo failas.
Po-sąlyga	Nuskaitytas skaitmeninis vaizdas iš nurodyto naudotojo failo.
Pagrindinis scenarijus	Iš naudotojo nurodyto failo nuskaitomas skaitmeninis vaizdas, skirtas tolimesniam vaizdo segmentavimo procesui atlikti.
Alternatyvus scenarijus	Nenurodžius tinkamo vaizdo failo, tolimesnis vaizdo segmentavimo procesas nėra leidžiamas.

7 lentelė. Panaudojimo atvejis „Atlikti semantinį skaitmeninio vaizdo segmentavimą“

Panaudojimo atvejis	Atlikti semantinį skaitmeninio vaizdo segmentavimą.
Tikslas/Uždavinys	Atlikti semantinį segmentavimą pasirinktam naudotojo skaitmeninio vaizdo failui.
Dalyviai	Naudotojas.
Prieš sąlyga	Turi būti pasirinktas tinkamas kelyje matomo skaitmeninio vaizdo failas.
Sužadinimo sąlyga	Inicijuojamas vaizdo apdorojimo procesas pasirinktam naudotojo failui.
Po-sąlyga	Atliktas semantinio segmentavimo procesas ir pateiktas segmentavimo rezultatas.
Pagrindinis scenarijus	Pasirinktam naudotojo skaitmeninio vaizdo failui atliekamas semantinio segmentavimo procesas.
Alternatyvus scenarijus	Neinicijavus vaizdo segmentavimo proceso, tolimesnis vaizdo apdorojimas nevykdomas.

8 lentelė. Panaudojimo atvejis „Atlikti dinaminio vaizdo segmentavimą“

Panaudojimo atvejis	Atlikti dinaminio vaizdo segmentavimą.
Tikslas/Uždavinys	Atlikti vaizdo segmentavimą naudotojo nurodytam dinaminiam vaizdo failui.
Dalyviai	Naudotojas.
Prieš sąlyga	Pasirinktas tinkamas skaitmeninis dinaminio vaizdo failas.
Sužadinimo sąlyga	Naudotojas inicijuoja vaizdo segmentavimo procesą pasirinktam dinaminiam vaizdo failui.
Po-sąlyga	Atliktas semantinio segmentavimo procesas ir rezultatai išsaugoti atitinkamoje sistemos direktorijoje.
Pagrindinis scenarijus	Atlikus segmentavimą, naudotojas informuojamas apie sėkmingą proceso užbaigimą, o rezultatai išsaugomi.
Alternatyvus scenarijus	Nepasirinkus tinkamo dinaminio vaizdo failo, segmentavimo procesas nevykdomas.

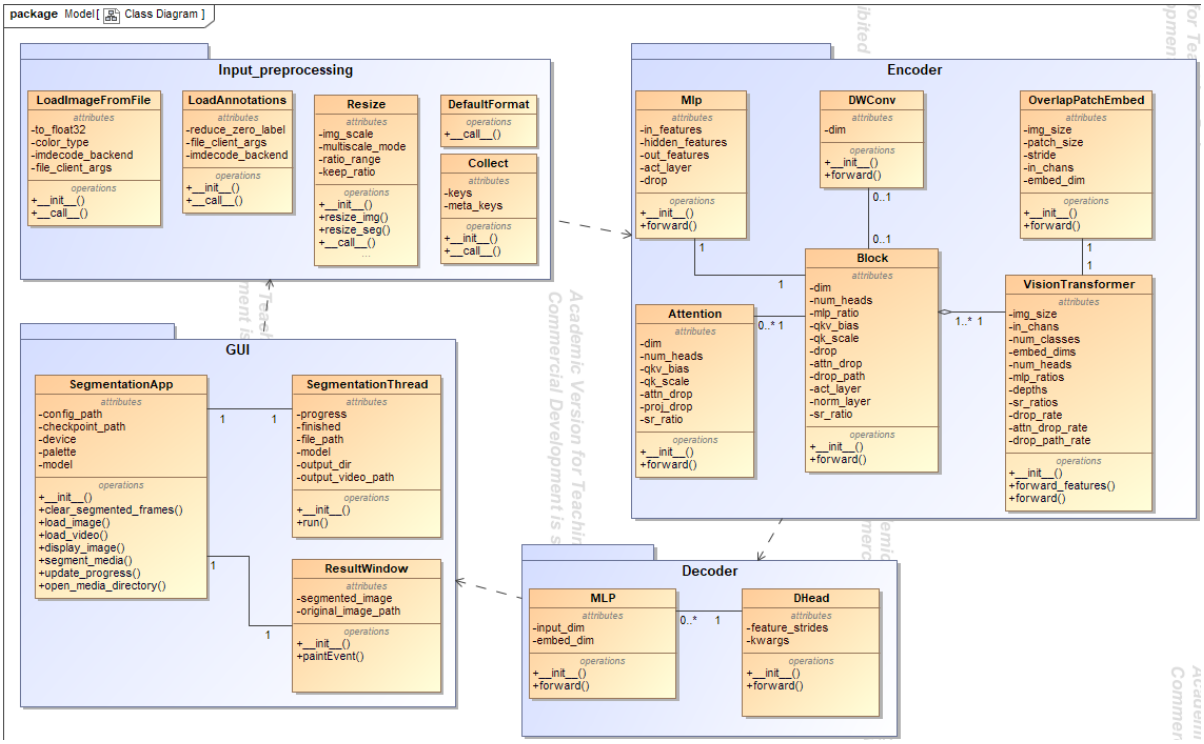
9 lentelė. Panaudojimo atvejis „Pateikti vaizdo segmentavimo rezultatą“

Panaudojimo atvejis	Pateikti vaizdo segmentavimo rezultatą
Tikslas/Uždavinys	Pateikti rezultatą po atlikto vaizdo segmentavimo proceso naudotojo pasirinktam failui.
Dalyviai	Naudotojas.
Prieš sąlyga	Pasirinktas vaizdo failas ir atliktas vaizdo segmentavimo procesas.
Sužadinimo sąlyga	Naudotojas inicijuoja vaizdo segmentavimo procesą pasirinktam skaitmeniniam vaizdo failui.
Po-sąlyga	Pateikiamas galutinis rezultatas po atlikto vaizdo segmentavimo proceso.
Pagrindinis scenarijus	Atlikus vaizdo segmentavimo procesą, naudotojui pateikiamas galutinis segmentavimo rezultatas.
Alternatyvus scenarijus	Neatlikus vaizdo segmentavimo proceso, naudotojas informuojamas klaidos pranešimu.

2.4. Statinis sistemos vaizdas

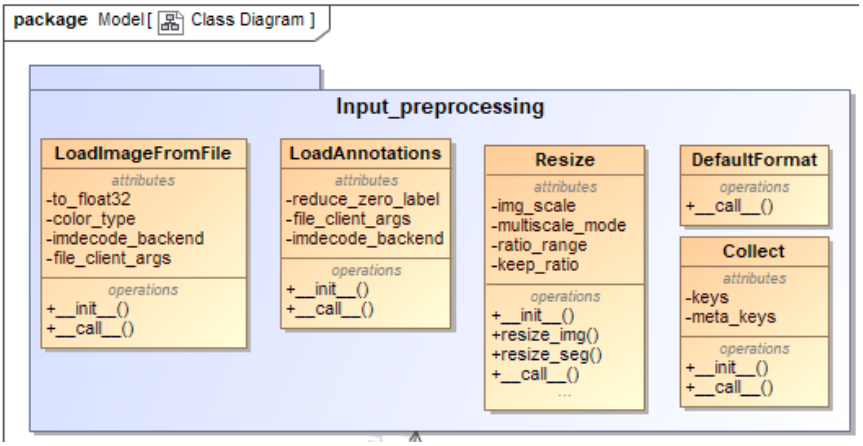
Šiame poskyryje esančiuose paveikslėliuose pateiktas bendras ir išskaidytas į skirtingus modulius sistemos architektūros vaizdas (žr. 9 pav.). Kompiuterinės regos kelyje matomo vaizdo segmentavimo sistemą sudaro keturi pagrindiniai moduliai:

- Įvesties duomenų apdorojimo modulis *Input preprocessing*;
- Kompiuterinės regos modelio kodavimo modulis *Encoder*;
- Kompiuterinės regos modelio dekodavimo modulis *Decoder*;
- Grafinės vartotojo sąsajos modulis *GUI*.



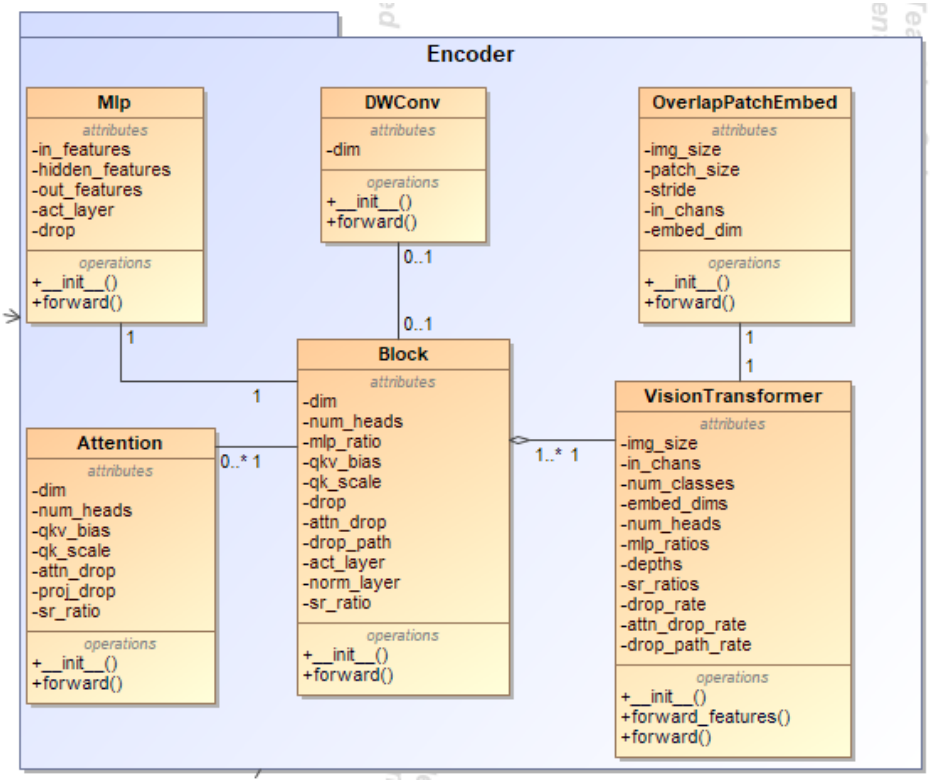
9 pav. Sistemos struktūra

10-ajame paveikslėlyje pavaizduotas sistemos modulis *Input preprocessing*, skirtas paruošti pradinius skaitmeninius įvesties vaizdus sistemos moduliui *Encoder*. Šis modulis atsako už įvesties vaizdų ir anotacijų užkrovimą, dydžio bei formato keitimą ir paruošimą tolimesniam vaizdo apdorojimo procesui tinkle.



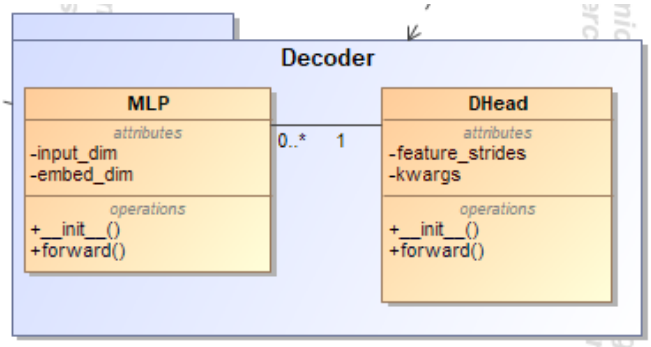
10 pav. Duomenų paruošimo modulis

11-ajame paveikslėlyje pavaizduotas pagrindinis sistemos *Encoder* modulis, skirtas išgauti įvairias vaizdo savybes iš įvesties vaizdų. Šiame komponente vykdomos įvairios operacijos įvesties vaizdams, o gauti duomenys perduodami į *Decoder* modulį, kuriame atliekama pikselių klasifikacija.



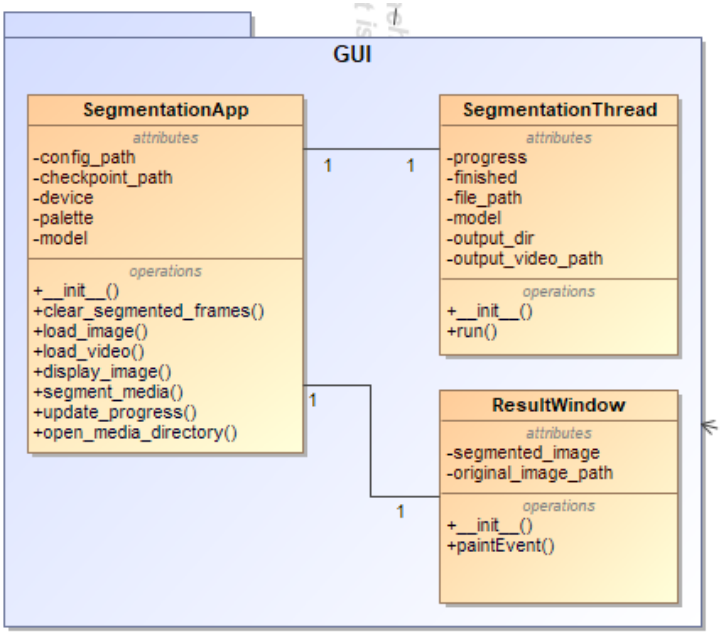
11 pav. Sistemos modulis *Encoder*

12-ajame paveikslėlyje pavaizduotas sistemos *Decoder* modulis. Šis modulis priima skirtingų dydžių savybių žemėlapius iš *Encoder* modulio, sujungia juos į vieną bendrą savybių rinkinį ir kiekvienam skaitmeniniame vaizde esančiam pikseliui priskiria atitinkamą klasę.



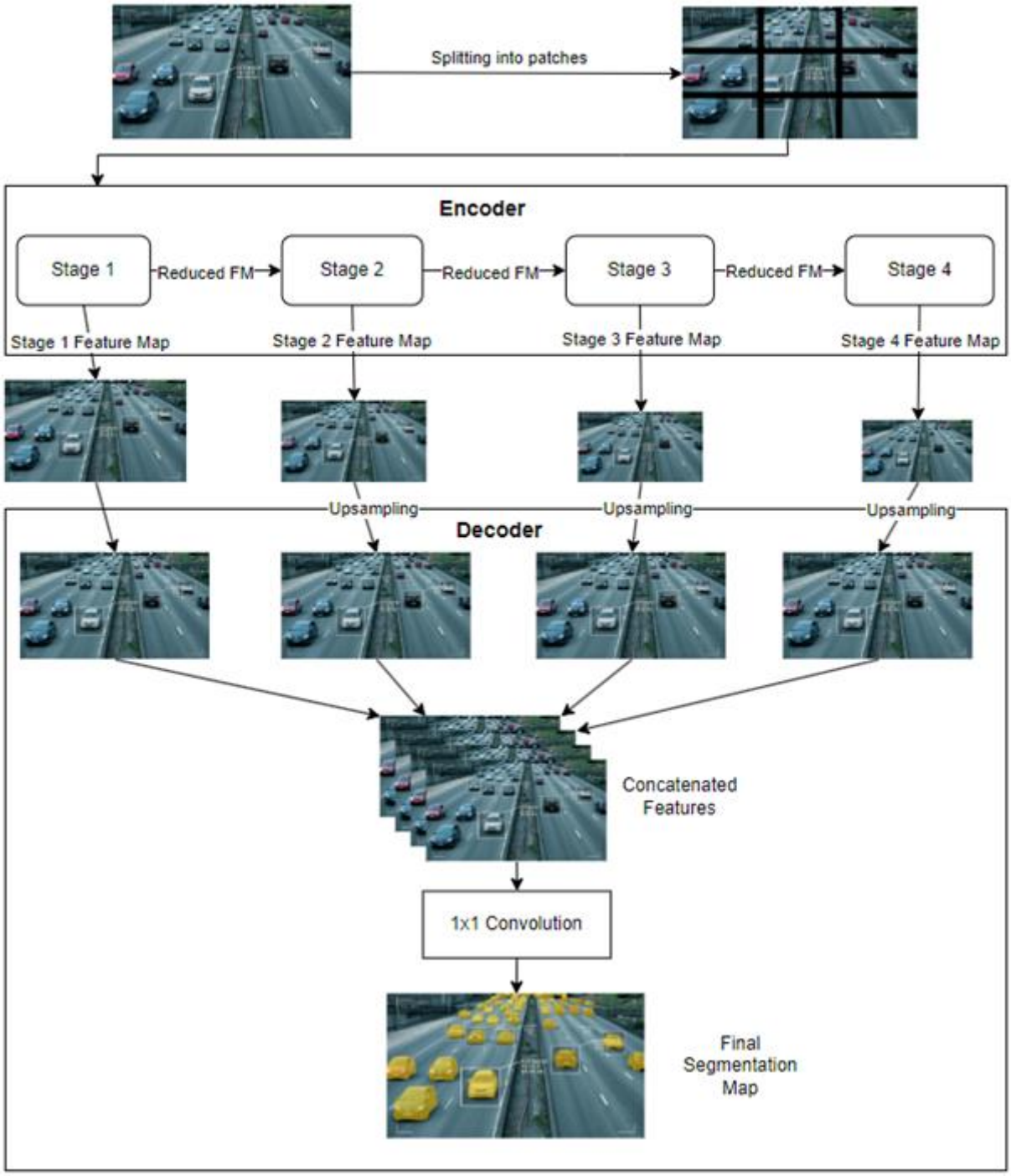
12 pav. Sistemos modulis *Decoder*

13-ajame paveikslėlyje pavaizduotas sistemos grafinės vartotojo sąsajos modulis *GUI*.



13 pav. Sistemos modulis *GUI*

14-ajame paveikslėlyje pavaizduotas kompiuterinės regos modelio veikimo principas. Proceso pradžioje įvesties vaizdas, prieš patenkant į sistemos kodavimo modulį, padalinamas į mažesnes dalis. Šios vaizdo dalys patenka į pirmąjį kodavimo modulio bloką *Stage 1*, kuriame vyksta vaizdo savybių išgavimas. Po kiekvieno etapo išgauti vaizdo savybių žemėlapiai perduodami į tolimesnius kodavimo modulio blokus, kur vyksta sekantis vaizdo savybių išgavimo procesas. Atlikus vaizdo savybių išgavimą, iš sistemos kodavimo modulio gaunami keturi skirtingų dydžių savybių žemėlapiai. Šie žemėlapiai perduodami į sistemos dekodavimo modulį, kuriame jie padidinami iki vienodo dydžio ir sujungiami į bendrą savybių visumą. Galiausiai, naudojant konvoliucinį sluoksnį, atliekamas pikselių klasifikavimas, priskiriant kiekvieną pikselį tam tikrai klasei ir spalvai. Proceso pabaigoje gaunamas galutinis vaizdo segmentavimo žemėlapis, identifikuojantis įvairius objektus skaitmeniniame vaizde.



14 pav. Kompiuterinės regos modelio veikimo principas

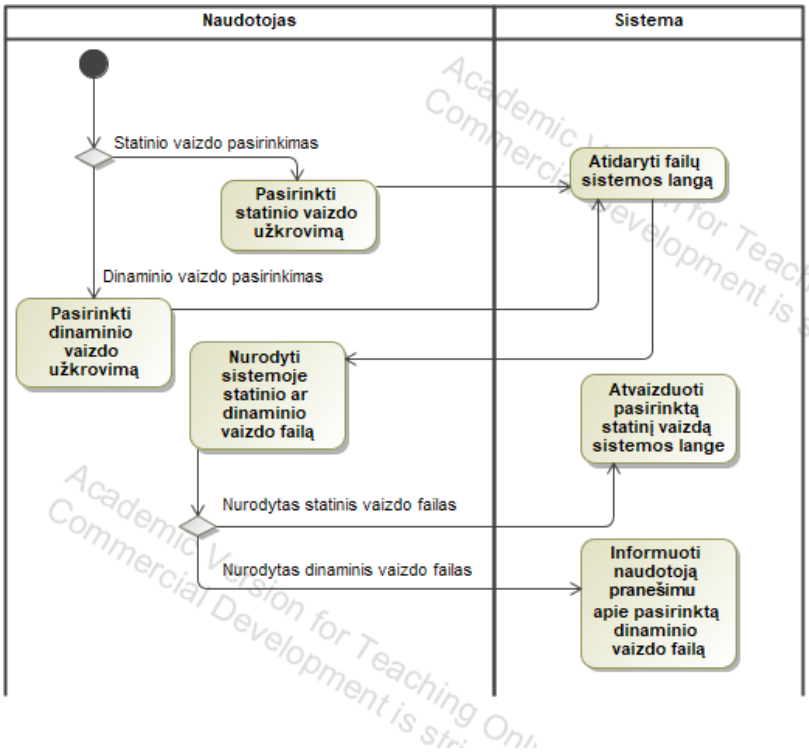
2.5. Dinaminis sistemos vaizdas

Šiame poskyryje pateiktose veiklos ir sekų diagramose pavaizduoti pagrindiniai kompiuterinės regos kelyje matomo vaizdo segmentavimo sistemos panaudojimo atvejai. Diagramos pateikiamos šiems panaudojimo atvejams:

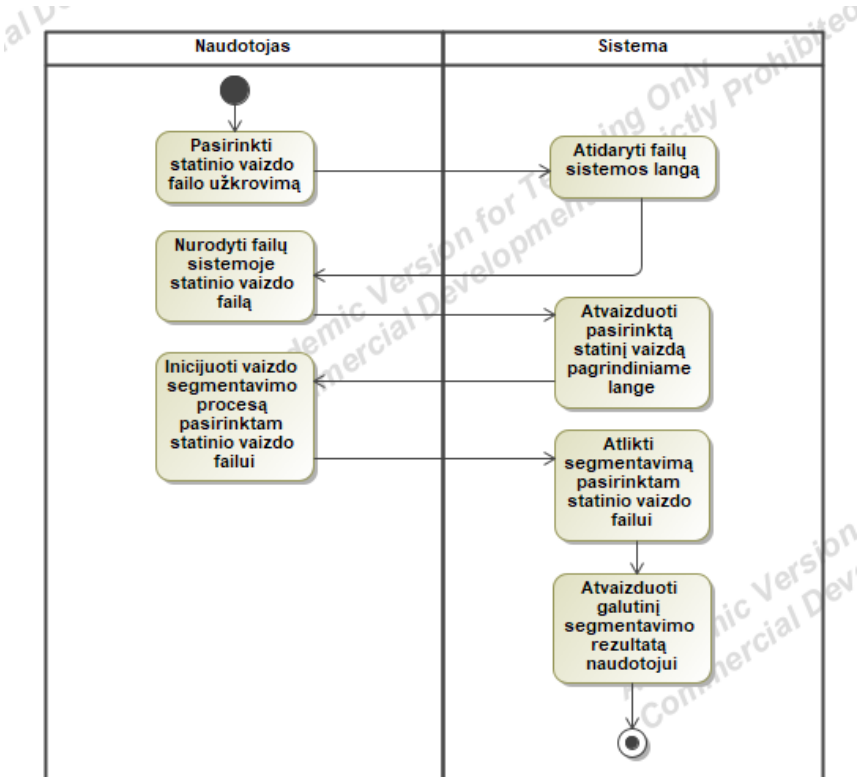
- Pasirinkti statinio ar dinaminio vaizdo failą;
- Atlikti statinio vaizdo segmentavimą;
- Atlikti dinaminio vaizdo segmentavimą.

Pateikti panaudojimo atvejai sudaro pagrindinį kompiuterinės regos kelyje matomo vaizdo segmentavimo sistemos funkcionalumą.

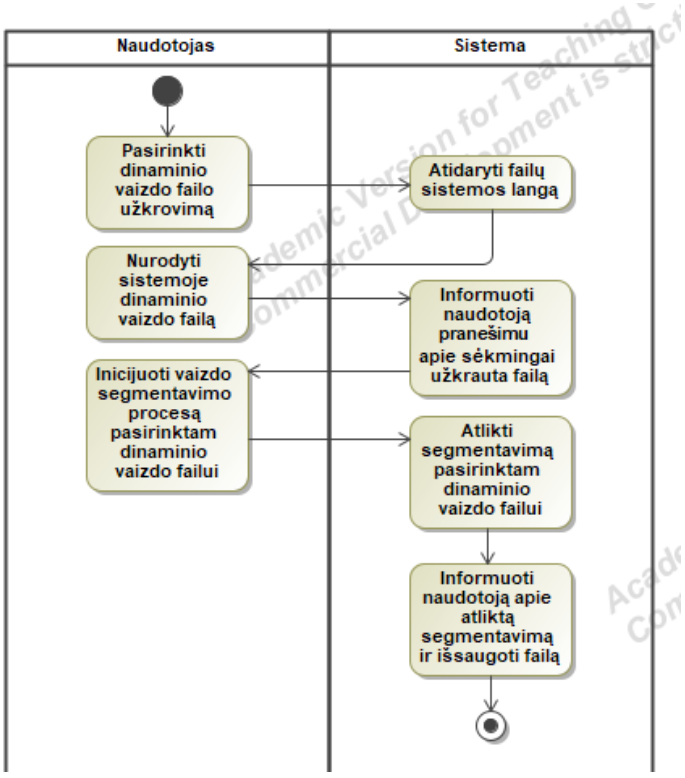
2.5.1. Veiklos diagramos



15 pav. Veiklos diagrama „Pasirinkti statinio ar dinaminio vaizdo failą“

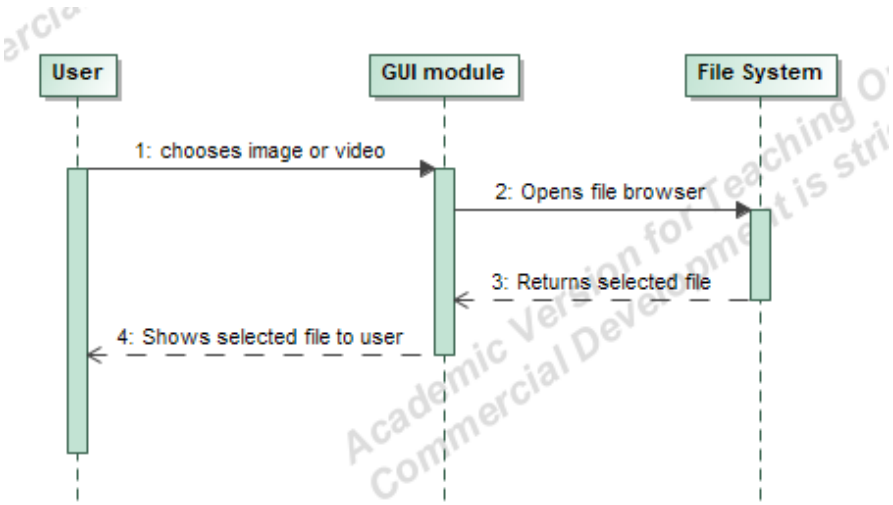


16 pav. Veiklos diagrama „Atlikti statinio vaizdo segmentavimą“

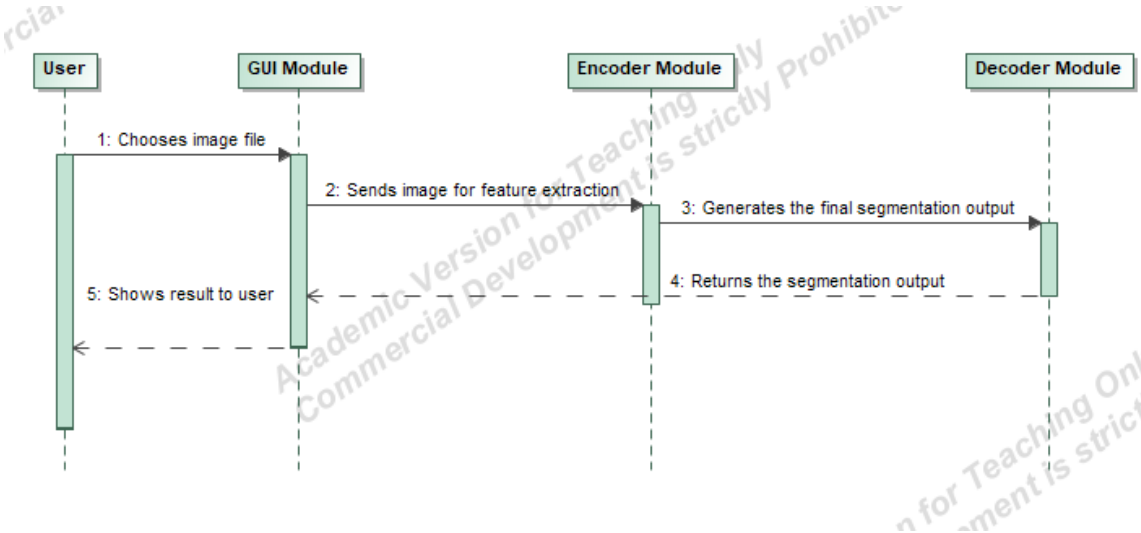


17 pav. Veiklos diagrama „Atlikti dinaminio vaizdo segmentavimą“

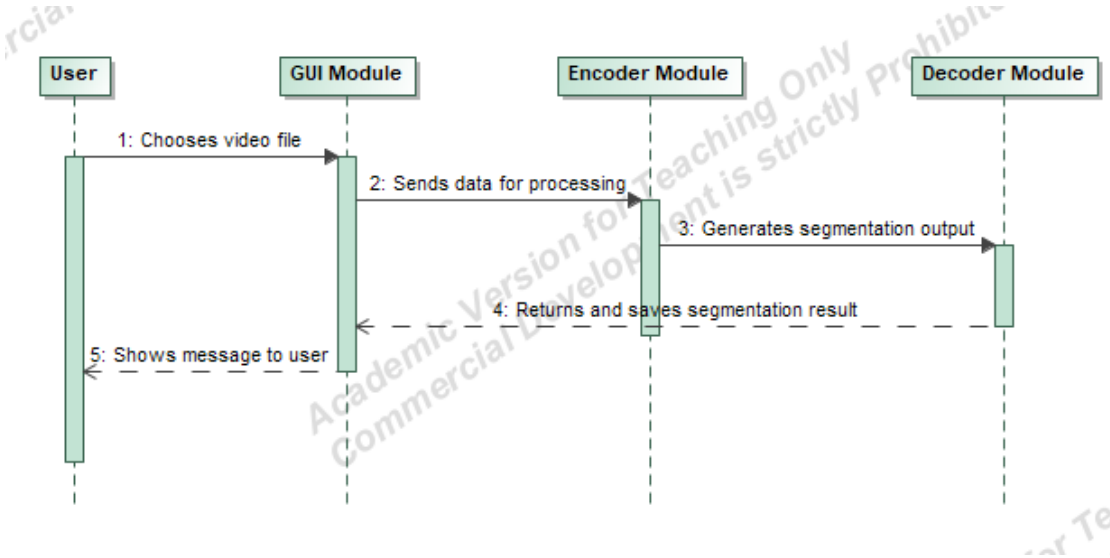
2.5.2. Sekų diagramos



18 pav. Sekų diagrama „Pasirinkti statinio ar dinaminio vaizdo failą“



19 pav. Sekų diagrama „Atlikti statinio vaizdo segmentavimą“

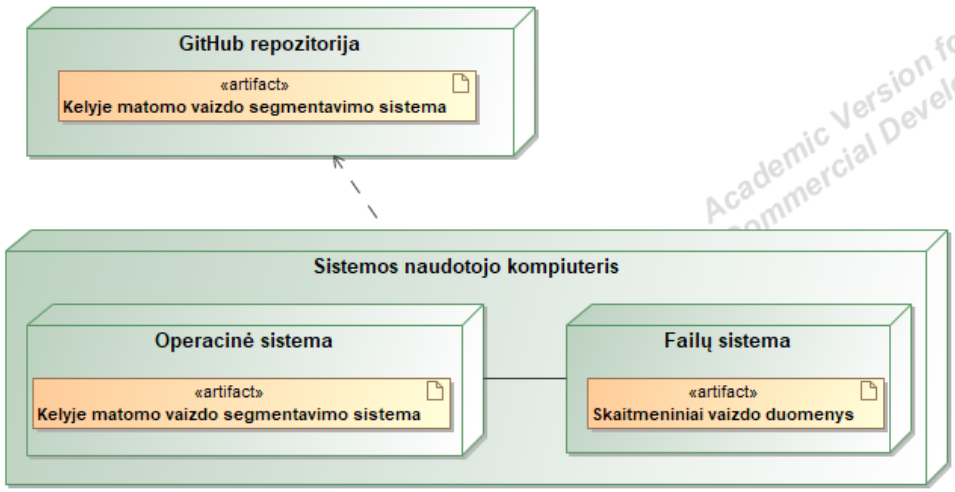


20 pav. Sekų diagrama „Atlikti dinaminio vaizdo segmentavimą“

2.6. Išdėstymo vaizdas

Kompiuterinės regos vaizdo segmentavimo sistema gali būti naudojama asmeniniame naudotojo kompiuteryje, importuojant *GitHub* repozitorijos turinį į sistemoje įdiegtą *PyCharm* integruotą kūrimo aplinką. Rekomenduojami reikalavimai vartotojui kuris naudosis sistema:

- *Ubuntu 20.04* operacinė sistema;
- 4 fizinių branduolių procesorius;
- 8 *GB RAM* operatyvioji atmintis;
- *Nvidia* grafinio apdorojimo procesorius su *CUDA* palaikymu.



21 pav. Išdėstymo diagrama

3. Eksperimentinė dalis

Eksperimentinėje dalyje aprašomos atliktos kompiuterinės regos modelio modifikacijos, eksperimentinės sąlygos ir gauti rezultatai. Skyriaus pabaigoje pateikiamas rezultatų apibendrinimas.

3.1. Duomenų rinkinys

Visi šiame skyriuje aprašyti eksperimentai ir sistemos apmokymai buvo atlikti naudojant plačiai žinomą *Cityscapes* duomenų rinkinį. Šis duomenų rinkinys naudojamas kompiuterinės regos kelyje matomo vaizdo segmentavimo modelių lyginamajai analizei atlikti. Naudojama duomenų rinkinio dalis sudaryta iš 5000 aukštos kokybės pikselių lygyje anotuotų miesto vaizdų, kurie buvo užfiksuoti 50 skirtingų Vokietijos miestų. Kiekvienas duomenų rinkinio skaitmeniniame vaizde esantis pikselis priskirtas vienai iš 19 skirtingų kategorijų, atspindinčių mieste esančius objektus ir aplinkos detales. Modelio apmokymui naudojamas duomenų rinkinys suskirstytas į tris skirtingas dalis:

- Mokymo dalis (2975 vaizdai): Šis duomenų poaibis naudojamas kompiuterinės regos modelio apmokymo metu;
- Validacijos dalis (500 vaizdų): Ši duomenų rinkinio dalis naudojama modelio apmokymo metu vertinti sistemos treniravimo progresą ir našumą;
- Testavimo dalis (1525 vaizdai): Šis duomenų poaibis naudojamas galutiniam modelio tikslumo įvertinimui, pateikiant sistemai dar nematytus skaitmeninius duomenis.

22 pav. Vaizdai iš *Cityscapes* duomenų rinkinio

Pateiktame 22-ajame paveikslėlyje yra keletas pavyzdžių iš *Cityscapes* duomenų rinkinio, kuris naudojamas modelio apmokymo ir testavimo metu. Pateiktuose paveikslėliuose galima matyti skirtingus objektus ir aplinką mieste. Visa informacija apie naudojamą duomenų rinkinį ir pilną klasių sąrašą, kuris buvo naudojamas kompiuterinės regos modelio apmokymo metu, yra viešai pasiekiama internete.

3.2. Kodavimo modulis

Sistemos kodavimo modulyje pagrindinis vaizdo savybių išgavimo tinklas paremtas *MixVisionTransformer* architektūros principu. Kiekvienoje aprašytoje sistemos modifikacijoje naudojamas tos pačios architektūros vaizdo savybių išgavimo tinklas, tik su skirtingais konfigūracijos parametrais. Šiame poskyryje pateikiama išsamesnė informacija apie vidinę tinklo struktūrą.

MixVisionTransformer vaizdo savybių išgavimo tinkle įvesties vaizdas pradiniame etape suskaidomas į mažesnius fragmentus, naudojant *Overlap Patch Embedding* modulį, kuris taiko 7x7 dydžio konvoliucinį filtrą su *Stride* parametro reikšme 4. Šis procesas leidžia įvesties vaizdą padalinti į persidengiančius fragmentus, kurių kiekvienas pirmajame kodavimo modulio etape paverčiamas į 64 kanalų savybių žemėlapį. Vėlesniuose kodavimo modulio etapuose taikomas 3x3 dydžio konvoliucinis filtras su *Stride* parametro reikšme 2. Naudojamas procesas leidžia sumažinti rezoliuciją ir padidinti kanalų skaičių. *MixVisionTransformer* tinklo *mit_b1* konfigūracijoje, keturiuose vaizdo savybių išgavimo etapuose, gaunami skirtingo kanalų skaičiaus savybių žemėlapiai:

- 1 etapas (c1): 64 kanalai;
- 2 etapas (c2): 128 kanalai;
- 3 etapas (c3): 320 kanalai;
- 4 etapas (c4): 512 kanalai.

10 lentelė. *MixVisionTransformer_b1* tinklo konfigūracijos parametrai

Parametras	**Reikšmė**	**Aprašymas**
Number of heads	[1, 2, 5, 8]	Dėmesio mechanizmų skaičius kiekviename etape.
MLP ratio	[4, 4, 4, 4]	Savybių išplėtimo koeficientas.
Depth	[2, 2, 2, 2]	Transformatoriaus blokų skaičius kiekviename etape.
Spatial reduction ratio	[8, 4, 2, 1]	Erdvės sumažinimo koeficientas.

Pateiktoje 10-oje lentelėje nurodyti *MixVisionTransformer* vaizdo savybių išskyrimo tinklo parametrai, naudojant *mit_b1* konfigūracijos nustatymus. Kiekviename vaizdo savybių išskyrimo etape naudojami du transformatoriaus blokai, kurie apdoroja įvesties duomenis ir išskiria reikšmingas savybes. Be to, kiekviename transformatoriaus bloke naudojamas skirtingas dėmesio mechanizmų skaičius. Kiekvienas tinklo etapas, dydžio sumažinimui taiko skirtingą erdvės sumažinimo koeficientą. Proceso pabaigoje išgautiems vaizdo savybių žemėlapiams atliekama normalizacija. Šiame vaizdo savybių išgavimo tinkle pradiniai etapai išgauna smulkias vaizdo detales, o vėlesni etapai – vis abstraktesnes ir aukštesnio lygio vaizdo savybes. Tolimesniame procese galutiniai vaizdo savybių žemėlapiai, gauti iš šio tinklo, perduodami į sistemos dekodavimo modulį, kuriame vykdoma tolimesnė pikselių klasifikacija.

3.3. Pirmoji sistemos modifikacija

Šiame poskyryje aprašoma modifikuota sistemos versija, kurioje naudojamas *MixVisionTransformer* vaizdo savybių išskyrimo tinklas su *mit_b3* konfigūracijos nustatymais. Atliekant šiame poskyryje aprašytą kompiuterinės regos modelio modifikaciją, sistemos apmokymui buvo naudojami tik *Cityscapes* duomenų rinkinio duomenys. Pagrindiniame vaizdo savybių išskyrimo tinkle eksperimento metu naudojami konfigūracijos nustatymai, kurie pateikti 11 lentelėje.

11 lentelė. *MixVisionTransformer_b3* tinklo konfigūracijos parametrai

Parametras	**Reikšmė**	**Aprašymas**
Number of heads	[1, 2, 5, 8]	Dėmesio mechanizmų skaičius kiekviename etape.
MLP ratio	[4, 4, 4, 4]	Savybių išplėtimo koeficientas.
Depth	[3, 4, 18, 3]	Transformatoriaus blokų skaičius kiekviename etape.
Spatial reduction ratio	[8, 4, 2, 1]	Erdvės sumažinimo koeficientas.

3.3.1. Modifikacijos aprašymas

Atlikus sistemos kodavimo modulyje vaizdo savybių išgavimo procesą, sistemos dekodavimo modulis gauna keturių skirtingų dydžių savybių žemėlapius. Tolimesniame etape šiame sistemos modifikacijos variante savybių žemėlapiai transformuojami į 256 matmenų erdvę, naudojant *MLP* modulius. Po šio proceso skirtingo dydžio savybių žemėlapiai transformuojami į vienodą rezoliuciją ir sujungiami į bendrą kanalų erdvę. Tolimesniame dekodavimo modulio procese pritaikomi *Depth wise separable convolution* sluoksniai. Galiausiai naudojant vieno pikselio konvoliucijos operaciją yra sukuriamas galutinis segmentavimo žemėlapis.

3.3.2. Eksperimentų sąlygos

12-oje lentelėje pateikti kompiuterinės regos modelio konfigūracijos nustatymai, naudoti sistemos apmokymo metu. Šioje sistemos modifikacijoje bendras kompiuterinės regos modelio parametrų skaičius tinkle siekia 44,61 milijono. Klasifikavimas vykdomas į 19 skirtingų klasių, identifikuojant įvairius objektus ir aplinką skaitmeniniuose vaizduose. Modelio mokymo procese buvo naudojami 512 × 512 pikselių dydžio įvesties vaizdai. Modelio mokymosi greičio parametras buvo nustatytas 0,00005. Eksperimento metu modelio parametrų atnaujinimui buvo naudojamas *AdamW* optimizavimo algoritmas. Sistemos mokymosi iteracijų skaičius nustatytas 160 000. Kompiuterinės regos modelio tikslumui vertinti pasirinkta vidutinė persidengimo metrika *mIoU*. Eksperimento metu naudojama *CrossEntropyLoss* nuostolio funkcija.

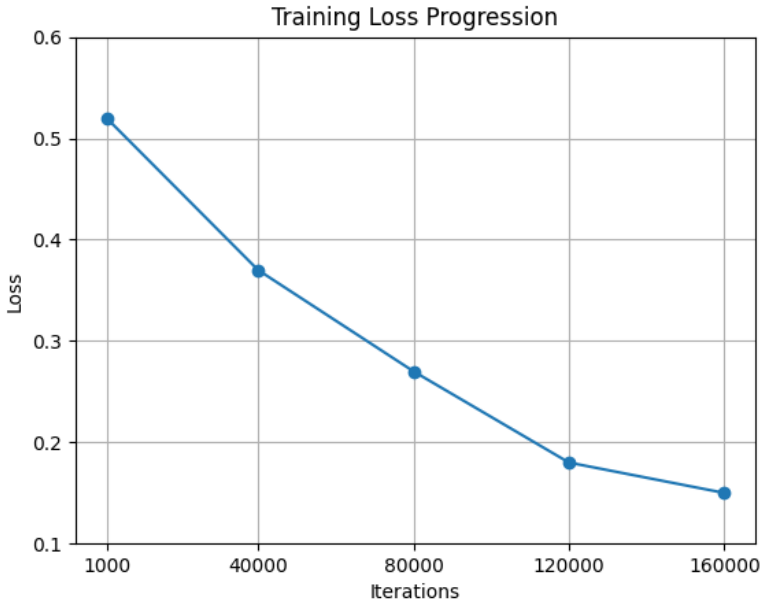
12 lentelė. Pirmosios sistemos modifikacijos konfigūracijos parametrai

Parametras	**Reikšmė**	**Aprašymas**
Model Size	44,61 mln.	Bendras modelio parametrų skaičius.
Training Image Size	512 x 512	Vaizdo dydis naudojamas modelio apmokymo metu.
Number of Classes	19	Klasifikuojamų klasių skaičius.
Learning Rate	0,00005	Modelio mokymosi greičio hiperparametras.
Optimizer	AdamW	Optimizavimo algoritmas modelio parametrų atnaujinimui.
Training Iterations	160 000	Modelio mokymo iteracijų skaičius.
Accuracy metric	mIoU	Modelio tikslumo vertinimo metrika.

Loss Function	CrossEntropyLoss	Nuostolio funkcija, matuojanti skirtumą tarp tikrų ir modelio prognozuojamų reikšmių.

3.3.3. Rezultatai

23-ajame paveikslėlyje pateiktas nuostolio funkcijos *CrossEntropyLoss* parametro pokytis, kompiuterinės regos modelio apmokymo metu. Iš pateiktos informacijos galime pastebėti, kad treniravimo proceso eigoje nuostolio funkcijos reikšmė nuosekliai mažėjo nuo pradinės 0,52 iki 0,15 ribos paskutinėse modelio treniravimo iteracijose.



23 pav. Pirmosios sistemos modifikacijos treniravimo procesas

13-oje lentelėje pateikiamos kiekvienos klasifikuojamos klasės tikslumo metrikos. Iš pateiktų duomenų galime matyti kiekvienos klasės tikslumą pagal persidengimo ir pikselių tikslumo metrikas. Pateikta pirmoji *IoU* metrika nurodo, kaip tiksliai modelio spėjamos pikselių reikšmės sutampa su tikrosiomis reikšmėmis. Trečiajame stulpelyje galime matyti, kiek tiksliai pikselių kiekvienoje klasėje modelis suklasifikavo.

13 lentelė. Pirmosios sistemos modifikacijos tikslumo metrikos

Klasė	**IoU (%)**	**Pikselių tikslumas (%)**
Road	96,80	98,63
Car	93,94	97,23
Sky	92,36	95,07
Vegetation	92,19	96,86
Building	90,84	96,66
Person	80,68	88,73
Bus	77,65	87,00
Sidewalk	76,82	86,02
Traffic Sign	76,28	82,97

Bicycle	75,85	86,91
Truck	74,18	80,17
Traffic Light	67,62	78,79
Train	66,04	79,17
Motorcycle	64,85	76,83
Terrain	60,89	65,92
Pole	60,15	68,87
Rider	58,70	77,80
Wall	50,81	56,66
Fence	48,41	53,78

3.3.4. Kokybinė analizė

Pateiktame 24-ajame paveikslėlyje yra kompiuterinės regos sistemos kokybiniai rezultatai. Nuotraukose yra atliekamas kelyje matomo vaizdo semantinis segmentavimas su šiame poskyryje aprašyta sistemos modifikacija. Kairėje nuotraukos pusėje galime matyti originalų vaizdą, viduryje yra pateikiamos tikrosios objektų ir aplinkos lokacijos, dešinėje nuotraukos pusėje yra modelio atlikti segmentavimo rezultatai.

24 pav. Pirmosios sistemos modifikacijos kokybiniai rezultatai

Iš pateiktų sistemos rezultatų galima pastebėti, kaip kompiuterinės regos modelis, turintis 44,61 milijono parametrų, geba aptikti įvairius objektus ir aplinką mieste. Pagrindiniai objektai ir aplinkos detalės, tokie kaip automobiliai, keliai, šaligatviai ir žmonės, trumpame nuotolyje aptinkami pakankamai geru tikslumu. Taip pat iš smulkių detalių galime pastebėti, kad kompiuterinės regos modelis kai kuriais atvejais trumpame nuotolyje stokoja tikslumo atskiriant kelio ir šaligatvio ribas.

Apibendrinant galima teigti, kad šiame poskyryje aprašytas sistemos modifikacijos eksperimentas parodė teigiamus segmentavimo rezultatus. Modelio apmokymo metu nuostolio funkcijos *CrossEntropyLoss* reikšmės nuosekliai mažėjo nuo pradinės 0,52 iki 0,15. Iš pateikto modelio mokymosi grafiko galima teigti, kad procesas buvo sėkmingas. Analizuojant tikslumo metrikas,

galima pastebėti, kad kompiuterinės regos modelis aptinka objektus ir aplinką, tokius kaip keliai, automobiliai, pastatai ir augmenija aukštesniu nei 90% tikslumu. Tačiau, kaip ir dauguma kitų semantinio segmentavimo modelių, sistema prasčiausiai segmentuoja tokias klases kaip tvoros, sienos ir stulpai. Šių klasių aptikimas yra vienas sudėtingiausių dėl esančių objektų įvairovės ir dydžio. Atlikta kokybinė modelio analizė patvirtiną, kad sistema geba sėkmingai identifikuoti pagrindinius objektus ir aplinką trumpoje distancijoje. Apibendrinant galima teigti, kad atliktas sistemos eksperimentas pademonstravo gerus vaizdo segmentavimo rezultatus. Modelis, turintis 44 milijonus parametrų, pasiekė 73,95 % vidutinės persidengimo metrikos tikslumo rezultatą.

3.4. Antroji sistemos modifikacija

Antroje sistemos modifikacijoje aprašoma sistemos versija, naudojanti *MixVisionTransformer* vaizdo savybių išskyrimo tinklą su *mit_b1* konfigūracijos nustatymais. Sistemos apmokymui buvo naudojami tik *Cityscapes* duomenų rinkinio skaitmeniniai duomenys. Pagrindinio vaizdo savybių išskyrimo tinklo konfigūracijos nustatymai pateikti 14 lentelėje.

14 lentelė. *MixVisionTransformer_b1* tinklo konfigūracijos parametrai

Parametras	**Reikšmė**	**Aprašymas**
Number of heads	[1, 2, 5, 8]	Dėmesio mechanizmų skaičius kiekviename etape.
MLP ratio	[4, 4, 4, 4]	Savybių išplėtimo koeficientas.
Depth	[2, 2, 2, 2]	Transformatoriaus blokų skaičius kiekviename etape.
Spatial reduction ratio	[8, 4, 2, 1]	Erdvės sumažinimo koeficientas.

3.4.1. Modifikacijos aprašymas

Šiame poskyryje aprašytos sistemos modifikacijoje dekodavimo modulis sukurtas efektyviai sujungti skirtingų matmenų vaizdo savybes, gautas iš vaizdo savybių išskyrimo tinklo. Proceso pradžioje gautos vaizdo savybės iš sistemos kodavimo modulio konvertuojamos į bendrą savybių erdvę, naudojant konvoliucinius sluoksnius, normalizaciją ir *ReLU* aktyvacijos funkciją tiesiškumo pašalinimui. Šis procesas užtikrina, kad iš sistemos kodavimo modulio gautos skirtingo dydžio vaizdo savybės būtų paruoštos tolimesniam sujungimo etapui. Kai vaizdo savybės yra sujungtos į bendrą savybių erdvę, pradedamas hierarchinis savybių sujungimo procesas. Proceso pradžioje mažiausios abstrakcijos lygio mažos rezoliucijos savybės padidinamos naudojant bilinearinę interpoliaciją, kad atitiktų aukštesnės rezoliucijos savybių žemėlapį, gautą iš sistemos kodavimo modulio. Tolimesniame etape naudojamas dėmesio mechanizmas, šių savybių efektyviam sujungimui. Atlikus hierarchinį savybių sujungimo procesą naudojant dėmesio mechanizmus, gautas aukštos raiškos savybių žemėlapis papildomai apdorojamas naudojant vieno pikselio konvoliucijos sluoksnį, siekiant sugeneruoti galutinį segmentavimo žemėlapį, kuriame kiekvienam skaitmeniniame vaizde esančiam pikseliui priskiriama atitinkama klasė.

3.4.2. Eksperimentų sąlygos

Pateiktoje 15-oje lentelėje yra kompiuterinės regos modelio konfigūracijos parametrai, naudoti sistemos apmokymo metu. Šioje kompiuterinės regos modelio modifikacijoje bendras tinklo parametrų skaičius siekia 17,16 milijono. Klasifikavimas atliekamas pagal *Cityscapes* duomenų rinkinio standartą, suskirstant į 19 skirtingų klasių ir aptinkant įvairius objektus bei aplinką skaitmeniniuose vaizduose. Modelio apmokymo metu buvo naudojamas 768x768 pikselių įvesties

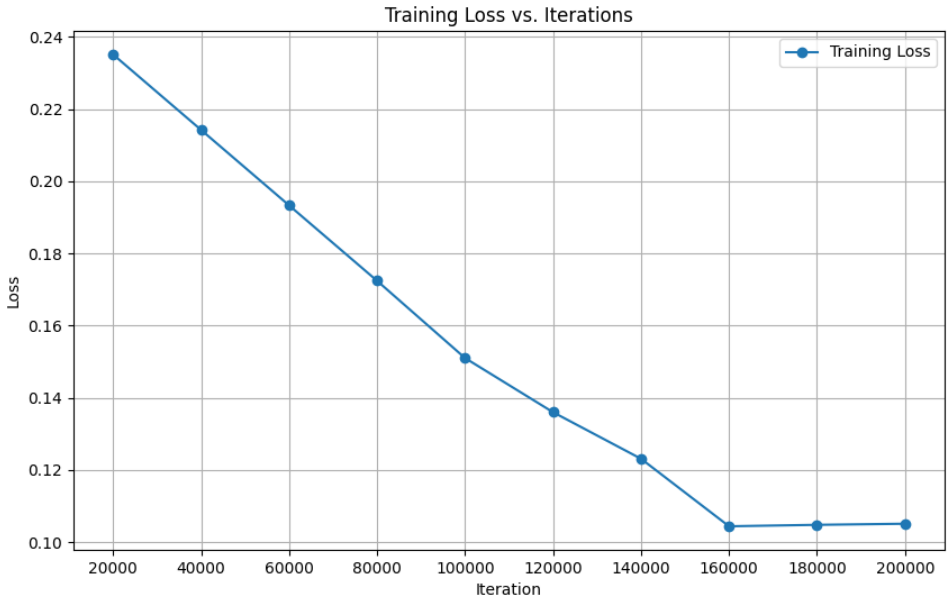
vaizdo dydis su nustatyta mokymosi greičio parametro reikšme 0,00005. Treniravimo metu naudojamas *AdamW* parametrų atnaujinimo algoritmas. Pirminio apmokymo etape modelio mokymosi iteracijų skaičius buvo nustatytas 160 000. Be to, atliktas papildomas bandymas apmokyti modelį iki 200 000 iteracijų, siekiant įvertinti, ar tolesnis mokymosi procesas išlieka efektyvus. Kompiuterinės regos modelio tikslumo įvertinimui pasirinkta naudoti vidutinė persidengimo metrika *mIoU*. Modelio treniravimo metu naudojama *CrossEntropyLoss* nuostolio funkcija.

15 lentelė. Antrosios sistemos modifikacijos konfigūracijos parametrai

Parametras	**Reikšmė**	**Aprašymas**
Model Size	17,16 mln.	Bendras modelio parametrų skaičius.
Training Image Size	768 x 768	Vaizdo dydis naudojamas modelio apmokymo metu.
Number of Classes	19	Klasifikuojamų klasių skaičius.
Learning Rate	0,00005	Modelio mokymosi greičio hiperparametras.
Optimizer	*AdamW*	Optimizacijos algoritmas modelio parametrų atnaujinimui.
Training Iterations	200 000	Modelio mokymosi iteracijų skaičius.
Accuracy metric	*mIoU*	Modelio tikslumo vertinimo metrika.
Loss Function	*CrossEntropyLoss*	Nuostolio funkcija, matuojanti skirtumą tarp tikrų ir modelio prognozuojamų reikšmių.

3.4.3. Rezultatai

25-ajame paveikslėlyje pavaizduotas kompiuterinės regos modelio nuostolio funkcijos *CrossEntropyLoss* reikšmės pokytis treniravimo proceso metu. Pateiktame grafike matyti, kaip sistemos nuostolio parametro reikšmės nuosekliai mažėjo viso treniravimo metu, pasiekdamos 160 000 iteracijų ribą. Paskutinėse modelio mokymosi iteracijose nuostolio parametras sumažėjo nuo 0,23 iki 0,11. Papildomas sistemos apmokymas iki 200 000 iteracijų nedavė teigiamo efekto, nes nuostolio parametro reikšmės išliko nepakitusios.



25 pav. Antrosios sistemos modifikacijos treniravimo procesas

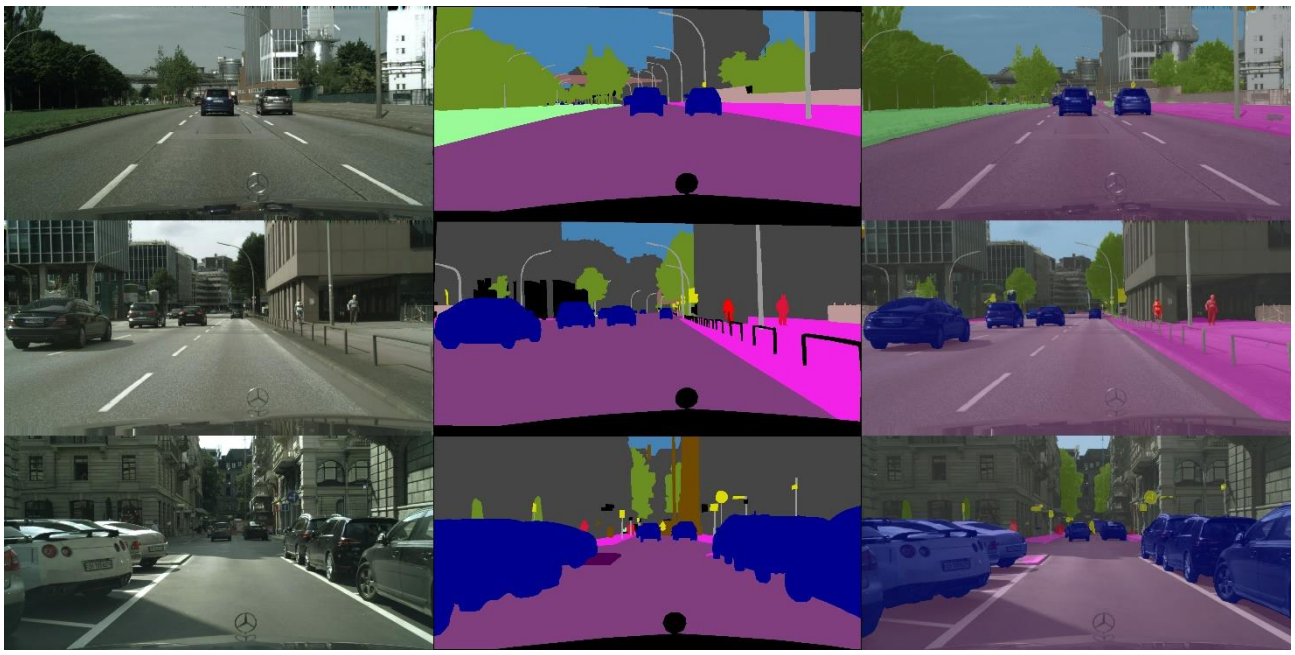
16-oje lentelėje pateikiamos kiekvienos klasifikuojamos klasės tikslumo metrikos. Lentelės antrajame stulpelyje pateikiami sistemos persidengimo metrikos *IoU* rezultatai, kurie rodo, kaip tiksliai modelio prognozuojamos pikselių reikšmės atitinka tikrąsias reikšmes. Trečiajame stulpelyje pateikiamos pikselių tikslumo metrikos, nurodančios kiek tiksliai modelis suklasifikavo pikselių kiekvienoje klasėje. Remiantis šiomis metrikomis, galima įvertinti modelio veikimo efektyvumą ir gebėjimą tiksliai atpažinti įvairius objektus bei miesto aplinką.

16 lentelė. Antrosios sistemos modifikacijos tikslumo metrikos

Klasė	**IoU (%)**	**Pikselių tikslumas (%)**
Road	97,93	98,93
Sky	94,54	98,15
Car	94,46	98,01
Vegetation	92,43	96,84
Building	92,16	96,43
Sidewalk	83,39	91,63
Person	80,90	91,60
Bus	80,04	85,36
Truck	79,17	84,65
Traffic Sign	78,83	85,16
Bicycle	76,71	87,72
Traffic Light	70,19	81,64
Motorcycle	65,86	76,66
Train	64,97	69,36
Terrain	63,02	69,89
Pole	62,83	72,81
Fence	59,84	70,10
Rider	59,45	71,76
Wall	55,14	62,86

3.4.4. Kokybinė analizė

Pateiktame 26-ajame paveikslėlyje yra atliktos sistemos modifikacijos kokybiniai rezultatai. Nuotraukose yra atliekamas kelyje matomo vaizdo semantinis segmentavimas su šiame poskyryje aprašyta sistemos modifikacija. Kairėje nuotraukos pusėje galima matyti originalų vaizdą, viduryje yra pateikiamos tikrosios objektų ir aplinkos lokacijos, dešinėje pusėje yra modelio atlikti segmentavimo rezultatai.



26 pav. Antrosios sistemos modifikacijos kokybiniai rezultatai

Pateiktame 26-ajame paveikslėlyje galima matyti atliktos sistemos modifikacijos vaizdo segmentavimo rezultatus ir gebėjimą aptikti įvairius objektus ir aplinką mieste. Rezultatai yra gauti su kompiuterinės regos modeliu kuris turi 17,16 milijono parametrų. Iš pateiktų segmentavimo rezultatų galima pastebėti, kaip sistema geba aukštu tikslumu aptikti įvairius objektus ir aplinkos detales skaitmeniniuose vaizduose. Poskyryje aprašyta kompiuterinės regos modelio modifikacija turi du kartus mažiau parametrų nei pirmoji sistemos versija, tačiau pasiekia aukštesnį 76,41 % vidutinės persidengimo metrikos tikslumo rezultatą. Galima teigti, kad šis kompiuterinės regos algoritmas yra efektyvesnis ir tikslesnis. Šios kompiuterinės regos modelio modifikacijos apmokymo metu nuostolio funkcijos *CrossEntropyLoss* reikšmės nuosekliai mažėjo viso treniravimo proceso metu, kol pasiekė 0,11 ribą. Iš pateikto modelio apmokymo grafiko galima matyti, kad sistemos apmokymas iki 160 000 iteracijų ribos buvo efektyvus. Pateikti sistemos kokybiniai rezultatai rodo, kad sistema geba pakankamai tiksliai aptikti pagrindinius objektus ir aplinkos detales mieste. Ši kompiuterinės regos modelio modifikacija aptinka kelią ir automobilius atitinkamai 97,93 % ir 94,46 % tikslumu. Panašiai kaip ir dauguma kitų kompiuterinės regos modelių, ši sistema labiausiai stokoja tikslumo aptinkant tokius objektus kaip stulpai, tvoros ar sienos. Visumoje galima teigti, kad poskyryje pateikta kompiuterinės regos algoritmo modifikacija yra pakankamai efektyvi ir tiksli aptinkant įvairius objektus ir aplinką. Ši kompiuterinės regos modelio modifikacija, turinti 17,16 milijono parametrų, pasiekia 76,41 % tikslumą naudojant Cityscapes duomenų rinkinį.

3.5. Trečioji sistemos modifikacija

Trečioje sistemos modifikacijoje aprašoma kompiuterinės regos modelio versija, naudojanti *MixVisionTransformer* vaizdo savybių išskyrimo tinklą su *mit_b4* konfigūracijos nustatymais. Sistemos apmokymui buvo naudojami tik Cityscapes duomenų rinkinio duomenys. Pagrindinio vaizdo savybių išskyrimo tinklo konfigūracijos nustatymai pateikiami 17 lentelėje.

17 lentelė. *MixVisionTransformer_b4* tinklo konfigūracijos parametrai

Parametras	**Reikšmė**	**Aprašymas**
Number of heads	[1, 2, 5, 8]	Dėmesio mechanizmų skaičius kiekviename etape.
MLP ratio	[4, 4, 4, 4]	Savybių išplėtimo koeficientas.
Depth	[3, 8, 27, 3]	Transformatoriaus blokų skaičius kiekviename etape.
Spatial reduction ratio	[8, 4, 2, 1]	Erdvės sumažinimo koeficientas.

3.5.1. Modifikacijos aprašymas

Šioje kompiuterinės regos modelio modifikacijoje naudojamas dekodavimo modulis su dėmesio mechanizmais, siekiant efektyviai sujungti skirtingas vaizdo savybes, gautas iš pagrindinio vaizdo savybių išskyrimo tinklo. Atliktoje dekodavimo modulio modifikacijoje vaizdo savybės iš sistemos kodavimo modulio konvertuojamos į vienodą 768 savybių erdvę, naudojant 1×1 konvoliucinius sluoksnius, grupinę normalizaciją ir *ReLU* aktyvacijos funkciją. Taip siekiama užtikrinti, kad visi vaizdo savybių žemėlapiai turėtų vienodą kanalų skaičių ir būtų tarpusavyje palyginami tolimesniame procese. Sistemos dekodavimo modulio modifikacijoje, hierarchiniame vaizdo savybių sujungimo etape, žemiausios rezoliucijos vaizdo savybės bilinearinės interpoliacijos būdu padidinamos iki aukštesnės rezoliucijos, kurios vėliau naudojamos sekančiame vaizdo savybių išskyrimo tinklo etape. Šioje dekodavimo modulio modifikacijoje taip pat naudojami dėmesio mechanizmai. Pagrindiniai atliktos modifikacijos komponentai atlieka šias funkcijas:

- Vaizdo savybių konvertavimą į bendrą 768 savybių erdvę;
- Hierarchinį vaizdo savybių sujungimą naudojant dėmesio mechanizmus;
- Sujungtų vaizdo savybių apdorojimą taikant konvoliucinius sluoksnius;
- Galutinę vaizdo pikselių klasifikaciją, generuojant segmentavimo žemėlapį.

Šiame poskyryje aprašyta dekodavimo modulio modifikacija leidžia efektyviai sujungti įvairių lygių vaizdo savybes, atpažinti svarbias vaizdo sritis ir pasiekti aukštą segmentavimo tikslumą.

3.5.2. Eksperimentų sąlygos

Šio skyrelio 18-oje lentelėje pateikti kompiuterinės regos modelio konfigūracijos nustatymai, naudoti sistemos apmokymo metu. Paskutinėje atliktoje sistemos modifikacijoje bendras tinklo parametrų skaičius viršija 95 milijonus. Eksperimento metu klasifikavimas vykdomas taip pat, kaip ir prieš tai aprašytuose eksperimentuose į 19 skirtingų klasių, laikantis *Cityscapes* duomenų rinkinio standarto. Kompiuterinės regos modelis buvo apmokytas naudojant 512×512 pikselių įvesties vaizdus. Mokymosi greičio hiperparametras nustatytas 0,00005. Sistemos treniravimui buvo naudojamas *AdamW* optimizavimo algoritmas. Sistemos apmokymui buvo nustatyta 160 000 iteracijų. Vaizdo segmentavimo modelio tikslumo įvertinimui naudojama *CrossEntropyLoss* nuostolio funkcija.

18 lentelė. Trečiosios sistemos modifikacijos konfigūracijos parametrai

Parametras	**Reikšmė**	**Aprašymas**
Model Size	95,29 mln.	Bendras modelio parametrų skaičius.
Training Image Size	512 x 512	Vaizdo dydis naudojamas modelio apmokymo metu.
Number of Classes	19	Klasifikuojamų klasių skaičius.
Learning Rate	0.00005	Modelio mokymosi greičio hiperparametras.

Optimizer	*AdamW*	Optimizacijos algoritmas modelio parametrų atnaujinimui.
Training Iterations	160 000	Modelio mokymosi iteracijų skaičius.
Accuracy metric	*mIoU*	Modelio tikslumo vertinimo metrika.
Loss Function	*CrossEntropyLoss*	Nuostolio funkcija, matuojanti skirtumą tarp tikrų ir modelio prognozuojamų reikšmių.

3.5.3. Rezultatai

27-ajame paveikslėlyje pavaizduotas sistemos modifikacijos nuostolio funkcijos *CrossEntropyLoss* reikšmės pokytis viso apmokymo proceso metu. Žvelgiant į pateiktą grafiką, galima matyti, kaip nuostolio funkcijos reikšmės mažėjo viso sistemos apmokymo metu, pasiekdamos žemiausią lygį ties 160 000 iteracijų riba. Galutinėse sistemos apmokymo iteracijose nuostolio funkcijos reikšmė stabilizavosi ties 0,10. Dėl to, tolimesnis treniravimas virš 160 000 iteracijų ribos nebuvo vykdomas.

27 pav. Trečiosios sistemos modifikacijos treniravimo procesas

Pateiktoje 19-oje lentelėje pateikiamos kiekvienos objektų ir aplinkos klasės tikslumo metrikos, gautos taikant šiame skyriuje aprašytą sistemos modifikaciją. Pirmame lentelės stulpelyje nurodyta klasifikuojama klasė, antrame stulpelyje procentinė persidengimo metrikos reikšmė ir trečiame nurodyta, kiek tiksliai pikselių kompiuterinės regos modelis tiksliai suklasifikavo.

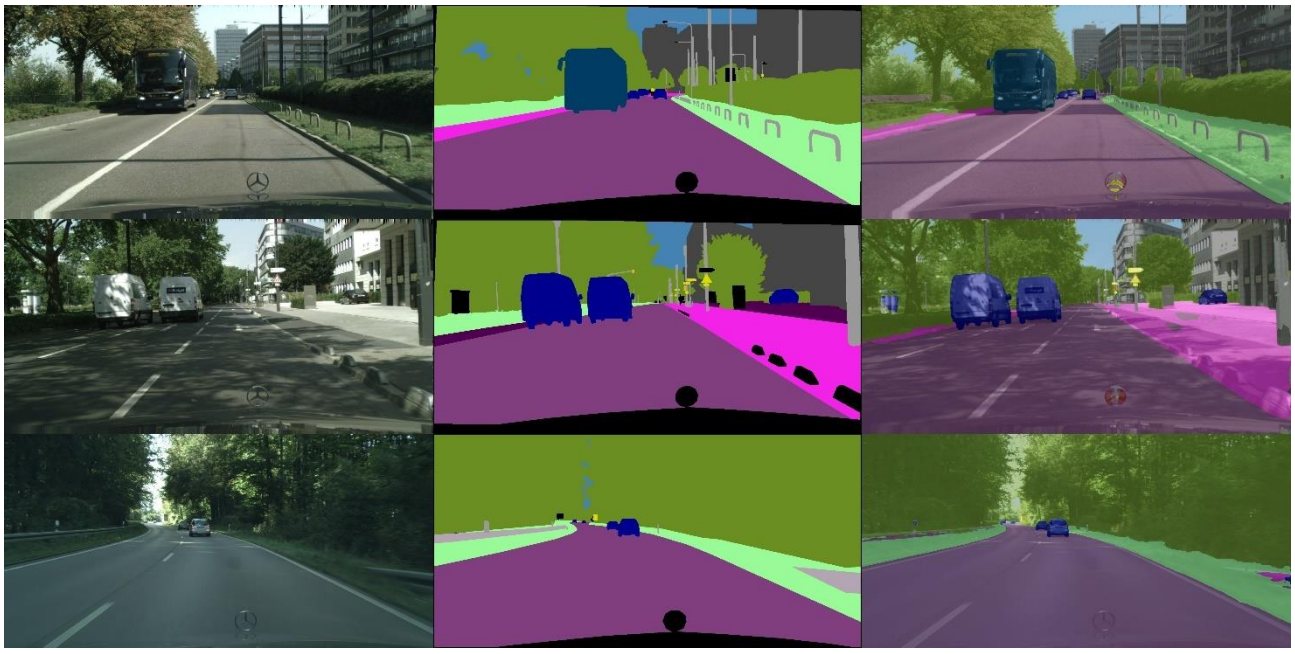
19 lentelė. Trečiosios sistemos modifikacijos tikslumo metrikos

Klasė	**IoU (%)**	**Pikselių tikslumas (%)**
Road	97,67	97,81
Car	94,28	97,52
Sky	92,63	96,79
Vegetation	92,42	96,65
Building	90,84	96,36

Bus	90,24	94,50
Train	85,34	89,59
Truck	85,11	93,91
Person	83,72	92,14
Sidewalk	82,75	90,44
Traffic sign	79,94	87,64
Bicycle	78,16	90,45
Motorcycle	73,21	86,68
Terrain	72,04	79,34
Pole	68,81	80,21
Traffic light	68,40	80,09
Rider	62,32	78,01
Fence	59,32	63,53
Wall	50,50	57,65

3.5.4. Kokybinė analizė

Pateiktame 28-ajame paveikslėlyje pavaizduoti trečiosios sistemos modifikacijos kokybiniai vaizdo segmentavimo rezultatai. Pirmame stulpelyje matomas tikrasis gatvės vaizdas, antrajame – tikrosios objektų ir aplinkos ribos, o trečiajame stulpelyje yra sistemos modifikacijos segmentavimo rezultatai, aptinkant įvairius objektus ir aplinkos detales skaitmeniniuose gatvės vaizduose.



28 pav. Trečiosios sistemos modifikacijos kokybiniai rezultatai

Pateiktuose sistemos segmentavimo rezultatuose matyti, kaip kompiuterinės regos vaizdo segmentavimo modelis, turintis daugiau nei 95 milijonus parametrų, geba identifikuoti įvairius objektus ir aplinkos detales gatvės vaizduose. Trečiosios sistemos modifikacijos rezultatai rodo, kad sistema geba pakankamai tiksliai aptikti pagrindinius objektus ir aplinkos detales kelyje. Ši sistemos modifikacija automobilius aptinka didesniu nei 97 % tikslumu, o kelią – didesniu nei 94 % tikslumu.

Trečioji, didžiausia kompiuterinės regos modelio modifikacija, turinti daugiau nei 95 milijonus parametrų, pasiekė 79,25 % vidutinės persidengimo metrikos tikslumo rezultatą. Iš sistemos apmokymo proceso rezultatų matyti, kad kompiuterinės regos modelio treniravimas buvo efektyvus. Šiai kompiuterinės regos modelio modifikacijai, kaip ir daugumai kitų kompiuterinės regos modelių, sunkiausia aptikti tokius objektus kaip tvoros ar sienos, kurių klasės pasiekia žemiausias tikslumo metrikas.

3.6. Sistemos modifikacijų palyginimas

Šiame poskyryje apibendrinami ir lyginami atliktų sistemos modifikacijų rezultatai, atsižvelgiant į kompiuterinės regos modelio dydį ir pasiektą vidutinės persidengimo metrikos tikslumo rezultatą. Pateiktoje 20-oje lentelėje yra kiekvienos kompiuterinės regos modelio modifikacijos tinklo dydžiai ir pasiekti vidutinės persidengimo metrikos tikslumo rezultatai.

20 lentelė. Sistemos modifikacijų palyginimas

Modifikacija	**Parametrų skaičius tinkle (mln.)**	**mIoU (%)**
Pirmoji	44,61	73,95
Antroji	17,16	76,41
Trečioji	95,29	79,25

Pirmoji atlikta sistemos modifikacija pasižymi vidutiniu modelio dydžiu. Kompiuterinės regos modelis sudarytas iš 44,61 milijono parametrų ir pasiekia 73,95 % vidutinės persidengimo metrikos tikslumo rezultatą. Iš visų šiame poskyryje atliktų sistemos eksperimentų, ši sistemos modifikacija turi vidutinį parametrų kiekį tinkle ir pasiekia mažiausią tikslumo rezultatą, atliekant kelyje matomo vaizdo semantinį segmentavimą.

Antroji sistemos modifikacija yra pati mažiausia, kompiuterinės regos modelis sudarytas iš 17,16 milijono parametrų ir pasiekia 76,41 % vidutinės persidengimo metrikos tikslumo rezultatą. Šioje sistemos modifikacijoje, dekodavimo modulyje yra naudojami dėmesio mechanizmai. Tai pakankamai geras kompromisas tarp sistemos efektyvumo ir tikslumo. Sąlyginai nedidelis parametrų kiekis tinkle, nedidelės atminties ir skaičiavimo resursų sąnaudos. Pagrindinių objektų ir aplinkos detalių aptikimas nedideliu atstumu pakankamai geras. Iš trūkumų galima paminėti, kad dėl nedidelio parametrų kiekio tinkle gali būti ribotas gebėjimas išmokti sudėtingas vaizdo detales. Tačiau įvertinant kompiuterinės regos modelio dydį, pagrindinių objektų ir aplinkos detalių aptikimas yra pakankamai aukštas.

Trečioji sistemos modifikacija yra pati didžiausia ir pasiekianti aukščiausią bendrą vidutinės persidengimo metrikos tikslumo rezultatą. Dėl didesnio parametrų kiekio tinkle ši sistemos modifikacija geba aptikti daugiau įvairių klasių aukštesniu tikslumu. Ši kompiuterinės regos modelio modifikacija sudaryta iš daugiau nei 95 milijonų parametrų tinkle. Modelis nors ir pasiekia aukščiausią vidutinės persidengimo metrikos tikslumą su *Cityscapes* duomenų rinkiniu, tačiau šis modelis turi ir pakankamai dideles atminties ir skaičiavimo resursų sąnaudas.

21 lentelė. Sistemos modifikacijų tikslumas skirtingose klasėse

Klasė	1 modifikacija	2 modifikacija	3 modifikacija
Road	96,80 %	97,93 %	97,67 %
Car	93,94 %	94,46 %	94,28 %
Building	90,84 %	92,16 %	90,84 %
Person	80,68 %	80,90 %	83,72 %
Sidewalk	76,82 %	83,39 %	82,75 %

Pateiktoje 21-oje lentelėje yra kiekvienos atliktos sistemos modifikacijos pasiekiamos tikslumo metrikos, identifikuojant įvairius objektus ir aplinkos detales skirtingose objektų ir aplinkos klasėse. Iš lentelės duomenų galima pastebėti, kad antroji sistemos modifikacija, naudojanti dekodavimo modulyje dėmesio mechanizmus, yra pakankamai efektyvus vaizdo segmentavimo algoritmas. Atsižvelgiant į esamą modelio dydį, kuris sudarytas iš 17 milijonų parametrų, ši modifikacija su *Cityscapes* duomenų rinkiniu pasiekia bendrą 76,41 % vidutinės persidengimo metrikos tikslumo rezultatą. Iš pateiktų duomenų matyti, kad sukurtas kompiuterinės regos algoritmas lenkia kitas sistemos modifikacijas beveik visose objektų ir aplinkos aptikimo klasėse. Geresnį tikslumą pasiekia tik trečioji sistemos modifikacija klasėje „*Person*“, turinti keturis kartus daugiau parametrų tinkle. Antroje sistemos modifikacijoje automobiliai aptinkami didesniu nei 94 % tikslumu, o keliai – didesniu nei 97 % tikslumu. Palyginus visas tris sistemos modifikacijas, akivaizdu, kad trečioji pasiekia aukščiausią bendrą tikslumą, tačiau reikalauja ir daugiausiai skaičiavimo resursų. Antroji modifikacija, turinti 17 milijonų parametrų tinkle, pasiekia 76,41 % vidutinės persidengimo metrikos tikslumą, išsiskiria mažiausiu parametrų kiekiu tinkle ir nedidelėmis atminties bei skaičiavimo resursų sąnaudomis.

Pateiktoje 22-oje lentelėje yra sukurto kompiuterinės regos modelio antrosios modifikacijos palyginimas su rinkoje esančiais semantinio segmentavimo modeliais, kurie buvo testuoti su *Cityscapes* duomenų rinkiniu. Lentelėje pateikiami naudoti įvesties vaizdo dydžiai, parametrų skaičius tinkle, skaičiavimo sąnaudos ir vidutinės persidengimo metrikos tikslumo rezultatai.

22 lentelė. Sistemos palyginimas su kitais rinkoje esančiais modeliais

Modelis	Vaizdo dydis	Parametrų skaičius (mln.)	GFLOPs	mIoU (%)
CSFNet-1	1024 x 512	12,6	86,9	74,8
Sukurtas modelis	512 x 512	17,2	37,9	76,4
DSNet	2048 x 1024	37,5	226,6	82,0
DeepLabV3+	2048 x 1024	43,5	1444,6	79,6
HRNetV2 + OCR	2048 x 1024	70,3	1206,3	81,6

Iš pateiktų kompiuterinės regos modelių rezultatų lentelėje galima pastebėti, kad antroji sistemos modifikacija pasižymi pakankamai nedideliu parametrų skaičiumi tinkle ir nedidelėmis skaičiavimo resursų sąnaudomis, pasiekdama 76,4 % tikslumą su Cityscapes duomenų rinkiniu. Pateikti modeliai, tokie kaip *DeepLabV3+* ir *HRNetV2 + OCR*, pasiekia geresnį bendrą tikslumo rezultatą, bet pasižymi pakankamai didelėmis skaičiavimo resursų sąnaudomis. Sukurtos kompiuterinės regos sistemos rezultatai buvo pristatyti tarptautinėje konferencijoje *IVUS* 2025 metais ir publikuoti žurnale *MDPI Machines* (žr. 1 ir 2 priedus).

Išvados

1. Atlikus literatūros analizę nustatyta, kad vaizdo segmentavimas skirstomas į semantinės, objektinės ir panoptinės segmentacijos rūšis. Atsižvelgiant į skirtingas segmentavimo rūšis, nuspręsta realizuoti kelyje matomo vaizdo semantinio segmentavimo sistemą.
2. Atlikus naudojamų technologijų, architektūrų bei duomenų rinkinių analizę paaiškėjo, kad kompiuterinės regos srityje plačiausiai taikomos konvoliucinių neuroninių tinklų, transformatorių ir jų hibridinės architektūros. Kelyje matomo vaizdo modelių mokymui dažnai naudojami tokie duomenų rinkiniai kaip *Cityscapes, CamVid* ir *Mapillary Vistas*.
3. Projekto metu buvo sukurtas kompiuterinės regos kelyje matomo vaizdo semantinio segmentavimo modelis, paremtas transformatoriaus neuroninio tinklo architektūra. Sistemos apmokymui naudotas *Cityscapes* duomenų rinkinys.
4. Atlikus kompiuterinės regos modelio antrosios modifikacijos testavimą su *Cityscapes* duomenų rinkiniu, nustatyta, kad 17 milijonų parametrų turintis modelis pasiekė 76,41 % vidutinės persidengimo metrikos tikslumo rezultatą. Antroji sistemos modifikacija pasižymi mažiausiu parametrų skaičiumi tinkle, tačiau, atsižvelgiant į kompiuterinės regos modelio dydį, pasiekia pakankamai gerą tikslumą daugumoje objektų ir aplinkos klasių. Nors trečioji sistemos modifikacija pasižymi aukštesniu bendru tikslumu (79,25 %), dėl žymiai didesnių skaičiavimo išteklių poreikio antroji modifikacija vertinama kaip efektyvesnė pagal tikslumo ir resursų sąnaudų santykį.
5. Palyginus sukurtą kompiuterinės regos modelį su kitais semantinio segmentavimo sprendimais, nustatyta, kad antroji modifikacija, turinti nedidelį parametrų skaičių ir mažas skaičiavimo resursų sąnaudas, yra pakankamai efektyvus kelyje matomo vaizdo segmentavimo algoritmas. Nors tokie modeliai kaip *DeepLabV3+* ar *HRNetV2+OCR* pasižymi aukštesniu tikslumu, jų skaičiavimo resursų poreikis yra gerokai didesnis.

Literatūros sąrašas

1. A. Garcia, S. Orts and S. Oprea, "A Review on Deep Learning Techniques Applied to Semantic Segmentation", *arXiv,* 2017.
2. U. Sehar and M. L. Naseem, "How deep learning is empowering semantic segmentation: Traditional and deep learning techniques for semantic segmentation: A comparison", *Multimedia Tools and Applications,* vol. 81, no. 21, p. 30519–30544, 2022.
3. X. Zhou, W. Gong and W. Fu, "Application of deep learning in object detection", 2017.
4. C. Rasche, "Computer Vision: An Overview for Enthusiasts", 2021.
5. S. Yuheng and Y. Hao, "Image Segmentation Algorithms Overview", *arXiv,* 2017.
6. R. Muthukrishnan and M. Radha, "Edge detection techniques for image segmentation", *International Journal of Computer Science and Information Technology,* vol. 3, p. 259–267, 2011.
7. V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation", *arXiv,* 2016.
8. F. Sultana, A. Sufian and P. Dutta, "Evolution of Image Segmentation using Deep Convolutional Neural Network: A Survey", *arXiv,* 2020.
9. S. Minaee, Y. Boykov, F. Porikli and A. Plaza, "Image Segmentation Using Deep Learning: A Survey", *arXiv,* 2020.
10. Z. Alom, T. Taha, C. Yakopcic, S. Westberg and P. Sidike, "The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches", *arXiv,* 2018.
11. A. Vaswani, N. Shazeer, N. Parmar and J. Uszkoreit, "Attention is All You Need", *arXiv,* 2017.
12. Y. Bai, J. Mei, A. Yuille and C. Xie, "Are transformers more robust than CNNs?", *arXiv,* 2021.
13. A. Dosovitskiy, L. Beyer, A. Kolesnikov and D. Weissenborn, "An image is worth 16x16 words. transformers for image recognition at scale", *arXiv,* 2021.
14. A. Tao, K. Sapra and B. Catanzaro, "Hierarchical multi-scale attention for semantic segmentation", *arXiv,* 2020.
15. L. Yuan, Q. Hou, Z. Jiang and J. Feng, "VOLO: vision outlooker for visual recognition", *arXiv,* 2021.
16. A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet classification with deep convolutional neural networks", 2017.
17. M. Cordts, M. Omran, S. Ramos and T. Rehfeld, "The cityscapes dataset for semantic urban scene understanding", *arXiv,* 2016.
18. G. Neuhold, T. Ollmann, S. Bulo and P. Kontschieder, "The Mapillary Vistas dataset for semantic understanding of street scenes", 2017.

19. S. Richter, V. Vineet, S. Roth and V. Koltun, "Playing for data: ground truth from computer games", *arXiv,* 2016.
20. B. Zhou, H. Zhao, X. Puig, S. Fidler and A. Barriuso, "Scene parsing through ADE20K dataset", 2017.
21. K. Okarma, "Applications of computer vision in automation and robotics", *Applied Sciences,* vol. 10, no. 19, 2020.
22. Z. Rafique, "Improving efficiency of computer vision for autonomous vehicles", 2020.
23. NASA, "Mars 2020 Perseverance Rover", 2020. [interaktyvus]. Available: https://science.nasa.gov/mission/mars-2020-perseverance/. [žiūrėta 19 3 2024].
24. R. Marasinghe, T. Yigitcanlar, S. Mayere, T. Washington and M. Limb, "Computer Vision Applications for Urban Planning: A Systematic Review of Opportunities and Constraints", *Sustainable Cities and Society,* vol. 100, 2024.
25. O. Iparraguirre-Gil, "Computer Vision and Deep Learning based Road Monitoring towards a Connected, Cooperative and Automated Mobility", 2025.
26. C. W. Landsiedel, "Semantic Mapping for Autonomous Robots in Urban Environments", Munich, 2018.
27. J. Ni, Y. Chen, G. Tang, J. Shi, W. Cao and P. Shi, "Deep Learning-Based Scene Understanding for Autonomous Robots: A Survey", *Intelligence & Robotics,* 2023.
28. O. Morayo and S. Folorunsho, "Reinforcement Learning in Autonomous Navigation: Overcoming Challenges in Dynamic and Unstructured Environments", *Engineering Science & Technology Journal,* 2024.
29. A. Ndidiamaka, T. Gilbert and A. Plastropoulos, "Advancements in Learning-Based Navigation Systems for Robotic Applications in MRO Hangar: Review", *Sensors,* 2024.
30. K. Katona, H. A. Neamah and P. Korondi, "Obstacle Avoidance and Path Planning Methods for Autonomous Navigation of Mobile Robot", *Sensors,* 2024.
31. J. Janai, F. Güney, A. Behl and A. Geiger, "Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art", *Foundations and Trends in Computer Graphics and Vision,* 2020.
32. M. V. Conde, "An Embarrassingly Pragmatic Introduction to Vision-based Autonomous Robots: Applications, Datasets and State of the Art", *arXiv,* 2021.
33. H. R. M. Pelikan, S. Reeves and M. N. Cantarutti, "Encountering Autonomous Robots on Public Streets", in *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI '24),* 2024.
34. S. Buckeridge, P. Carreno-Medrano, A. Cosgun, E. Croft and W. P. Chan, "Autonomous Social Robot Navigation in Unknown Urban Environments Using Semantic Segmentation", *arXiv,* 2022.
35. R. Siegwart, I. R. Nourbakhsh and D. Scaramuzza, "Introduction to Autonomous Mobile Robots, Cambridge: MIT Press", 2011.

36. E. Dilek and M. Dener, "Computer Vision Applications in Intelligent Transportation Systems: A Survey", *Sensors,* 2023.

37. D. Upadhyay, D. K. Upadhay, R. Singh, D. Mishra and V. Khatri, "Robotic Vision: Advancements in Computer Vision for Autonomous Systems", *Propulsion Technology Journal,* 2023.

38. S. Fernandes, D. Duseja and R. Muthalagu, "Application of image processing techniques for autonomous cars", 2021.

39. Q. Sellat, S. Bisoy, R. Priyadarshini, A. Vidyarthi, S. Kautish and D. R. Barik, "Intelligent Semantic Segmentation for Self-Driving Vehicles Using Deep Learning", *Computational Intelligence and Neuroscience,* pp. 1-10, 2022.

40. Y. Xiang, C. Xie, A. Mousavian and D. Fox, "Unseen object instance segmentation for robotic environments", *arXiv,* 2021.

41. E. Goceri, "Challenges and recent solutions for image segmentation in the era of deep learning", 2019.

42. U. Sehar and M. Luqman, "How deep learning is empowering semantic segmentation", 2022.

43. Y. Wei, H. Hu, Z. Xie and Z. Zhang, "Contrastive learning rivals masked image modeling in fine-tuning via feature distillation", *arXiv,* 2022.

Priedai

1 priedas. Sistemos publikacija IVUS

Urban Road Segmentation with Transformers

Bartas Lisauskas[1], Rytis Maskeliūnas[1]

[1]Kaunas University of Technology, Faculty of Informatics, Studentų St.50, Kaunas, Lithuania

Abstract

This paper introduces a transformer-based computer vision system for segmenting different urban road scenes. Detection and understanding of objects in the environment is a critical task for autonomous vehicles or advanced self-driving robots. The approach integrates a MiT transformer-based backbone network for feature extraction with a decoder that incorporates CNN depthwise separable convolution layers, to efficiently fuse features and reduce computational cost. The system detects and separates different objects and environments into 19 semantic classes, as defined by the Cityscapes dataset. The computer vision model consists of 44.61 million parameters and reaches the mean intersection over union of the 73.95% accuracy metric with the Cityscapes dataset. The results gathered demonstrate the good ability of the model to detect different objects and environments in urban road scenes. The proposed computer vision system approach demonstrates the balance between good segmentation accuracy and efficient network structure for more reliable autonomous solutions in complex urban environments.

Keywords

Computer vision, Deep learning, Image processing, Neural networks, Semantic segmentation

1. Introduction

Computer vision is a field of artificial intelligence that teaches computers to understand the world as humans see it. Using deep learning models and digital image data, systems can accurately identify and classify objects in various road scene environments, and based on that information, autonomous systems can make further decisions. Today, computer vision systems help automate processes in various domains. As with any rapidly evolving field, it is increasingly challenging to keep up with the latest knowledge. Computer vision systems use neural networks to perform image processing tasks. One such task is to extract useful information from digital images. Neural networks are applied to object detection, classification, and segmentation tasks. From an engineering perspective, the goal of computer vision is to develop autonomous systems that can perform tasks that human beings do, and often do so more quickly and efficiently.

Image segmentation is one of the most important digital image processing techniques and has been widely used in the automotive and robotics industries. Road scene segmentation is a critical problem when computer vision systems are deployed for autonomous driving, pedestrian detection, and traffic monitoring. In autonomous vehicles, the quality and reliability of computer vision systems are very important for the safety of the driver and other road users. A precise understanding of traffic participants or obstacles is essential to prevent potential accidents, and accurate object detection in environments with many traffic users is a fundamental requirement to achieve safe, efficient, and reliable autonomous driving. However, developing a system that can reach high precision remains a challenging task.

In recent years, the need for accurate segmentation of the road scene has grown significantly, especially within the automotive and robotics industries. Autonomous vehicles are highly dependent on accurate understanding of the surroundings, pedestrians, and obstacles to ensure safe navigation. According to a 2021 investigation conducted in the US by the National Highway Traffic Safety Administration, the results showed that approximately 94% of all car accidents are due to human error [1].

The automotive and robotics industries have a potential market for different computer vision systems. The automotive industry is rapidly moving towards the realization of autonomous vehicle technologies.

IVUS2025: International Conference Information Society and University Studies, May 15, Kaunas, Lithuania
bartas.lisauskas@ktu.edu (B. Lisauskas); rytis.maskeliunas@ktu.lt (R. Maskeliūnas)
0000-0000-0000-0000 (B. Lisauskas); 0000-0002-2809-2213 (R. Maskeliūnas)

The integration of computer vision systems in self-driving cars is expected to continue to drive substantial market growth. According to Forbes, in 2022, investments in the automotive industry to achieve autonomous driving technologies exceeded 200 billion dollars. In addition, data from 2021 reveal that more than 80 companies in the United States are actively testing more than 1400 different autonomous vehicles. This information underscores the potential and critical importance of computer vision systems in improving the efficiency and safety of autonomous transportation. Autonomous driving systems not only aim to reduce human driving errors, but can also offer benefits such as more convenient travel, increased mobility, lower operating costs, increased road safety, and reduced ecological footprints [2, 3].

This paper presents a computer vision system that is specifically developed for the segmentation task of different road scene environments. The system performs a detailed image analysis by identifying and classifying various objects and environments on the road and generating segmentation masks for each detected element in the digital view.

2. Semantic Segmentation task and Modern Approaches

Computer vision is a field of artificial intelligence that focuses on the recognition and processing of different objects and environments with digital images. Semantic segmentation is the process of partitioning a digital image into different segments based on shared visual patterns, and it is a fundamental and challenging task in the field of computer vision. During the segmentation process, each pixel in the image is identified, and pixels with similar visual characteristics are grouped into the same classes to differentiate objects from each other and from the background environment. During the past decade, the rapid evolution of deep learning and neural network architectures has substantially improved performance in these tasks, and deep learning models have become indispensable for extracting and processing complex information. For deep learning models, architecture and data set are critical components, and the choice and quality of these directly affect the accuracy of the systems in the execution of computer vision tasks [4].

The path to modern semantic segmentation was introduced with early convolutional neural networks (CNNs). In 1989, French scientist Yann LeCun proposed one of the first CNN architectures, it was called LeNet-5. The computer vision model was built for the handwritten digit recognition task. This work not only demonstrated the effectiveness of using CNN architectures for pattern recognition but also paved the way for their application to more complex tasks, such as image classification, object detection, or image segmentation. During the past decade, different CNN architectures have become the backbone of the neural network of different computer vision applications due to their ability to learn hierarchical representations directly from raw pixel data [5].

Although CNN architectures have been dominating the computer vision field for many years, a proposed transformer architecture approach in 2017 by Ashish Vaswani changed the situation [6]. The Vision Transformers (ViTs) architecture demonstrated a new approach to the important feature extraction process by processing images as sequences of patches and using self-attention mechanisms. This demonstrated approach allowed models to capture long-range dependencies and global context without the constraints that CNN architecture had with fixed-size convolutional filters. Compared with many years used CNN architectures, transformer architectures dynamically focus on the most relevant parts of the image, enhancing robustness to variations in scaling, rotation, or occlusions. In addition, transformer-based models can scale better with larger amounts of data, making them a competitive alternative for semantic segmentation tasks, where global scene understanding is critical [7, 8].

Modern semantic segmentation systems now often use these transformer-based methods with conventional CNN techniques. Earlier demonstrated proposals, such as Fully Convolutional Networks (FCNs), used a different approach by replacing fully connected layers with convolutional layers to enable end-to-end pixel-wise prediction. When implementing systems in FCNs, encoder-decoder architectures such as U-net introduced skip connections that recover fine-grained spatial details that were lost during the downsampling process, while models such as DeepLabV3+ further advanced the field by using dilated convolutions and spatial pyramid pooling to capture multiscale context effectively [9, 10].

In addition, recent published approaches have combined the strengths of CNNs and transformers in hybrid architectures. These models use the efficiency of CNNs for local feature extraction together with the transformer's ability to integrate global contextual information using self-attention mechanisms. Using this type of combination allows for a more accurate segmentation process, particularly in complex scenes such as urban road environments where precise object boundaries and contextual information are essential [11].

In summary, the evolution from the proposed first CNN architecture model LeNet-5 to modern transformer-based architectures today shows significant progress in the field of computer vision, performing different computer vision tasks. Today's state-of-the-art models can achieve high-accuracy results with different image segmentation tasks. Transformer-based architecture models has an advantage with improved global context understanding by using self-attention transformer mechanisms. By integrating of these advantages it is possible to reach good results in the semantic segmentation task, which leads to more accurate and efficient computer vision systems.

3. Methods and Dataset

In this section all the details are provided about the structure of architecture, configuration settings for training and evaluating phases of the computer vision model. In addition, more detailed information is provided about the data set that was used for the training and evaluation steps.

3.1. Dataset

For the experiments which were made with the computer vision model, only the Cityscapes dataset was used. This data set is widely known for benchmarking purposes for many computer vision models in image segmentation tasks. The whole data set consists of 5000 high-resolution images, which were recorded in 50 different cities across Germany. Each pixel in each image is annotated in one of 19 semantic classes such as road, vehicles, pedestrians, traffic signs, or sidewalks. The Cityscapes dataset has three different data splits:

- **Training set:** 2975 images that are used in the training process of a computer vision model.
- **Validation set:** 500 images that are used during the training process to evaluate the model and monitor performance during the training phase.
- **Test set:** 1525 images that are used for the final evaluation process phase, to gather information on the accuracy metrics of the trained model with unseen images.

Figure 1: Sample images from Cityscapes dataset across different Germany cities.

In Figure 1, we can see the images provided from the Cityscapes dataset with different landscape scenes in urban areas. All information on the data set used in the training and evaluation phases of the model and the full list of semantic classes with annotation examples is publicly available at https://www.cityscapes-dataset.com/.

3.2. Configuration Details

The proposed computer vision transformer-based road scene segmentation model is implemented using the mmsegmentation framework codebase. The model consists of an encoder-decoder architecture and uses MiT b3 configuration settings in the encoder module [8]. The system encoder module is pre-trained on ImageNet-1k dataset for better feature extraction task. To improve model accuracy and generalization, in the training process, data augmentation techniques were applied using only a Cityscapes dataset. In the training process, additional functionality was used for random horizontal flipping, cropping, and scaling. The crop size of 512 x 512 pixels was chosen during the training phase. At inference time, the entire image testing strategy was used to generate segmentation predictions. The computer vision model was trained using an AdamW optimization algorithm with an initial learning rate set to 0.00005. Due to GPU resource constraints, a batch size of 1 image was used in the training process. The training schedule was set for 160 000 iterations. The performance of the model was evaluated using the widely used mean intersection-over-union (mIoU) accuracy metric.

3.3. Decoder

The decoder in the system receives four different sets of feature maps with different resolution from the encoder after the feature extraction process. Later these feature maps are projected into a 256 lower dimensional embedding space by using multi-layer perceptron modules. After this projection, different features are resized to the same spatial resolution and concatenated along the channel dimension. The combined feature map is processed by a depth-wise separable convolutional layer. This layer first performs a spatial convolution on each channel independently and then applies a pointwise 1x1 convolution to fuse the information across different channels. Using this approach, an efficient multiscale feature fusion process is possible. Finally, in the decoder part, a dropout layer is applied for better regularization, and a 1x1 convolutional layer produces the final output of the segmentation map.

4. Results

This section provides results of the computer vision road scene segmentation model. In the following, information is provided on the results of the model training process, global accuracy, and finer per-class accuracy metrics. Qualitative visual results are also provided to better understand how the system is capable of detecting different objects and environments with the road scene images.

4.1. Training Process

During the computer vision model training process, the training schedule was set to 160 000 iterations. No further training process was conducted beyond 160 000 iterations line. Model training was performed using AdamW optimizer and CrossEntropyLoss function. In Figure 2, the training phase graph is provided to better understand how the model reduced the loss parameter during the training cycle.

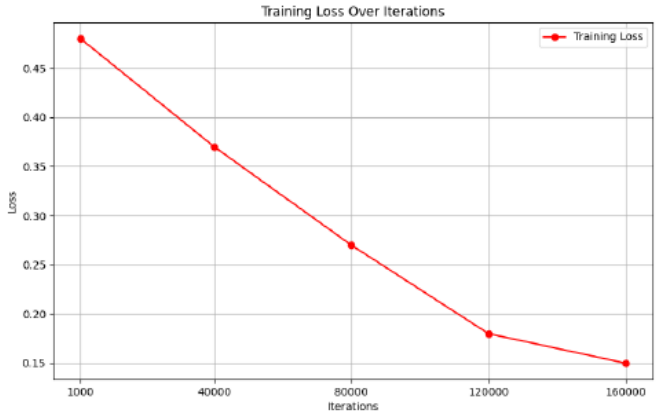


Figure 2: Graph with model training loss values over 160 000 iterations.

The information provided in Figure 2 is gathered from the training logs. It is clearly visible that the model loss parameter was steadily decreasing throughout the training cycle, up to the 160.000 iteration line. From the initial loss of 2.12 to the final phases, where it reached approximately 0.15 at the end of the training cycle. The model best performance was reached at the 160 000 iterations line.

4.2. Per-Class Performance

The computer vision model segments scenes into 19 classes according to the standard of the cityscape data set. In the table below, an intersection over union and accuracy metrics is provided for each class.

Table 1
Per-Class Performance Metrics

Class	IoU (%)	Accuracy (%)
Road	96.80	98.63
Car	93.94	97.23
Sky	92.36	95.07
Vegetation	92.19	96.86
Building	90.84	96.66
Person	80.68	88.73
Bus	77.65	87.00
Sidewalk	76.82	86.02
Traffic Sign	76.28	82.97
Bicycle	75.85	86.91
Truck	74.18	80.17
Traffic Light	67.62	78.79
Train	66.04	79.17
Motorcycle	64.85	76.83
Terrain	60.89	65.92
Pole	60.15	68.87
Rider	58.70	77.80
Wall	50.81	56.66
Fence	48.41	53.78

From the data provided in Table 1, it is clear that the system can detect objects and elements of the

environment, such as roads, cars, the sky, vegetation, and buildings, with a high precision above 90%. Road and car classes reach the highest accuracy metrics. Still, there is room for improvement in the future with other vehicle classes that have medium accuracy results. In addition, classes with lowest accuracy metrics are the most difficult to detect for many computer vision models. It is a challenging task when a distinction is needed between the wall and the fence, but the accuracy can be improved with additional data or the modified architecture with the bigger neural network.

4.3. Global Metrics

To obtain the overall performance accuracy results, the computer vision model was evaluated on the Cityscapes validation set with 500 images to get the global accuracy results. The following global accuracy metrics were collected:

- **Mean IoU (mIoU):** 73.95%
- **Mean Accuracy (mAcc):** 81.79%
- **Overall Accuracy (aAcc):** 95.07%

The metrics provided demonstrate that the computer vision model, which has 44.61 million parameters, is capable of reaching a global 73.95% mean intersection over the union accuracy metric. This metric reflects the average overlap between the predicted segments and the ground-truth values in all classes. It is used primarily to evaluate the performance of many computer vision models. Based on the data provided, we can also see that the computer vision model reached the mean accuracy value (mAcc) of 81.79%. This metric represents the average classification accuracy per pixel for each class. In addition, the overall accuracy metric (aAcc) of 95.07% was reached. This metric reflects the ratio of correctly classified pixels to the total number of pixels in the Cityscapes validation set.

4.4. Qualitative Analysis

Figure 3 demonstrates an example of computer vision system capability to detect different objects and environment in the images of the road scene. In the provided figure, the original images are on the left side, and the results after the segmentation process are on the right side.

Figure 3: Example images with original one on the left, and segmented result on the right.

From this side-by-side images comparison, we can see that the system is capable to reach good accuracy when detecting different objects and environment on the road scene images in close distance. Objects or environments such as cars, roads, or pathways are detected with high accuracy. However, it is worth mentioning that some challenges remain in accurately segmenting distant objects, which leaves a potential direction for future improvements.

4.5. Computational complexity

As shown in Table 2, computer vision model with 44.61 million parameters scales from 41.84 GFLOPs at 512×512 resolution to 238.24 GFLOPs at 1024×1024 resolution, illustrating the trade-off between computational cost and input resolution.

Table 2
Computational Complexity at Different Input Resolutions

Input Resolution	GFLOPs	Parameters (M)
512 × 512	41.84	44.61
768 × 768	110.76	44.61
1024 × 1024	238.24	44.61

5. Conclusion

The experimental results demonstrate that this transformer-based approach for road scene segmentation achieves good accuracy results when detecting different objects and environment details with diverse urban scenes, while keeping the architecture relatively lightweight. With 44.61 million parameters in the network, the computer vision model can reach the accuracy metric of 73.95% mIoU with the Cityscapes validation set. The computer vision model uses the MiT transformer-based encoder as the backbone component for feature extraction, and the CNN-based decoder incorporating depthwise separable convolution layers to efficiently fuse features and reduce computational resources. The visual results provided show that the system can accurately detect major objects and environmental elements in a close distance. However, lower-accuracy classes with small or more distant objects indicate the future area for improvements, suggesting that additional training data or architecture modifications may enhance system performance in these challenging cases. To further improve segmentation accuracy in future work, the computer vision model can be trained on additional road scene datasets such as Berkeley Deep Drive, Mapillary Vistas, or CamVid, which together have tens of thousands of varied urban driving images. The current results were obtained using only the 2975 images from the Cityscapes training set, without any additional data. On the architecture side, it is possible to replace the current CNN-based decoder with a custom attention-based fusion module designed to preserve fine spatial details and capture long-range context. Early experiments indicate that an alternative approach can even use the smaller feature extraction transformer network, reducing overall model size while improving global segmentation accuracy results.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] Zohaib Rafique, Improving efficiency of computer vision for autonomous vehicles (2020). URL: http://rgdoi.net/10.13140/RG.2.2.11761.92009. doi:10.13140/RG.2.2.11761.92009.

[2] Q. Sellat, S. Bisoy, R. Priyadarshini, A. Vidyarthi, Intelligent semantic segmentation for self-driving vehicles using deep learning, 2022. URL: https://www.researchgate.net/publication/357907010_Intelligent_Semantic_Segmentation_for_Self-Driving_Vehicles_Using_Deep_Learning.

[3] Forbes, Autonomous vehicles and their impact on the economy, 2022. URL: https://www.forbes.com/councils/forbestechcouncil/2022/02/14/autonomous-vehicles-and-their-impact-on-the-economy/.

[4] X. Zhou, W. Gong, W. Fu, Application of deep learning in object detection, 2017. URL: https://www.researchgate.net/publication/318035834_Application_of_deep_learning_in_object_detection.
[5] F. Sultana, A. Sufian, P. Dutta, Evolution of image segmentation using deep convolutional neural network: A survey, Knowledge-Based Systems 201–202 (2020) 106062. doi:10.1016/j.knosys.2020.106062.
[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2017. arXiv:1706.03762, [arXiv:cs.CL/1706.03762].
[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020).
[8] E. Xie, W. Wang, Z. Yu, X. Lei, P. Luo, Segformer: Simple and efficient design for semantic segmentation with transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 1202–1211.
[9] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, 2015, pp. 234–241.
[10] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2018, pp. 801–818.
[11] K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, W. Wu, Incorporating convolution designs into visual transformers, arXiv preprint arXiv:2103.11816 (2021).

2 priedas. Sistemos publikacija MDPI

machines

an Open Access Journal by MDPI

IMPACT FACTOR 2.1

CITESCORE 3.0

CERTIFICATE OF ACCEPTANCE

The certificate of acceptance for the manuscript (**machines-3584031**) titled:

Efficient Transformer-Based Road Scene Segmentation Approach with Attention-Guided Decoding for Memory-Constrained Systems

Authored by:

Bartas Lisauskas; Rytis Maskeliunas

was accepted in *Machines* (ISSN 2075-1702) on 26 May 2025

MDPI

Basel, May 2025

Prof. Dr. Antonio J. Marques Cardoso
Editor-in-Chief

MDPI

Article

Efficient Transformer-Based Road Scene Segmentation Approach with Attention-Guided Decoding for Memory-Constrained Systems

Bartas Lisauskas [1,†,‡], **Rytis Maskeliunas** [1,*]

[1] Kaunas University of Technology, Faculty of Informatics, Software Engineering Department, Kaunas, Lithuania; bartas.lisauskas@ktu.edu, rytis.maskeliunas@ktu.lt
* Correspondence: rytis.maskeliunas@ktu.lt
† Current address: Affiliation.
‡ These authors contributed equally to this work.

Abstract: Accurate object detection and surroundings understanding are key requirements when applying computer vision systems in the automotive or robotics industries, with autonomous vehicles or self-driving robots. A precise understanding of road users or obstacles is essential to avoid potential accidents. Due to many objects and the diversity of the environment, the segmentation task of the road scene remains a challenging one. In our approach, a transformer-based backbone is employed for robust feature extraction in the encoder module. In addition, we have developed a custom decoder module in which we implemented attention-based fusion mechanisms to effectively combine features. The decoder modification is specifically designed to maintain fine spatial details and enhance global context understanding, setting our method apart from conventional approaches that typically use simple projection layers or standard query-based decoders. The implemented model consists of 17.2 million parameters and achieves competitive performance with a mean intersection over union (mIoU) of 76.41% metric on the Cityscapes validation set. The results gathered indicate the ability of the model to capture both global context and fine spatial details critical to the accurate segmentation of urban scenes. Furthermore, the lightweight design makes the approach suitable for deployment on memory-limited devices.

Keywords: Computer Vision; Deep Learning; Image processing; Neural networks; Semantic segmentation

Received:
Revised:
Accepted:
Published:

Citation: Lisauskas, B.; Maskeliunas, R. Efficient Transformer-Based Road Scene Segmentation Approach with Attention-Guided Decoding for Memory-Constrained Systems. *Machines* **2025**, *1*, 0. https://doi.org/

1. Introduction

Neural networks are used to perform image processing tasks in computer vision. One such task is to extract useful information from digital images. These networks are used to perform object detection, classification, and segmentation tasks. From an engineering perspective, the goal of computer vision is to create autonomous systems that can perform tasks that humans do, and in many cases, do so faster and more efficiently.

Autonomous driving is likely to be one of the revolutionary technologies that will have a very significant impact on people's daily lives in the future. Image segmentation systems provide autonomous cars with a view of the surrounding world and are critically important in achieving safe autonomous vehicle driving. According to a study conducted by the US National Highway Traffic Safety Administration, 94% of all traffic accidents are caused by human error [1]. The realization of autonomous car driving aims to solve this huge problem of car accidents. Since autonomous systems are programmed to drive

efficiently and safely, they reduce, and sometimes even eliminate, the need for human driving, thus eliminating the aforementioned human errors [2]. The automotive industry is a very promising area for developing computer vision solutions in autonomous vehicles. The more accurate the image segmentation process and the shorter the time it takes, the more accurately autonomous vehicles will understand their surroundings and be able to make safer decisions. The giving of autonomous vehicles a view of the world around them using computer vision solutions can provide many benefits, such as increased road safety, lower costs, more comfortable travel, greater mobility, or smaller ecological footprints [3].

With the rapid development of autonomous driving technology, an accurate visual understanding of the surroundings is crucial to ensure road safety and efficient autonomous vehicle navigation. Accurate detection and classification of objects such as pedestrians, vehicles, and traffic signs are essential because it directly impacts the safety and efficiency of autonomous driving systems. Image segmentation is one of the most important processes in digital image processing and has been widely used in the automotive industry and robotics in recent years. The field of computer vision, which is related to artificial intelligence, has made great progress in the past decade, and today's computer vision systems can recognize visual data faster than humans. In the field of computer vision, the semantic segmentation task remains one of the most challenging ones. The assignment of class labels to each pixel and the classification task at the pixel level is a key approach that enables vehicles to distinguish between different objects, roads, pathways, pedestrians and the remaining environmental elements of the road.

As computer vision systems are widely applied in the automotive industry, the same methods can be equally applied to a broader spectrum of autonomous systems in the robotics or aerospace industries. For example, during the NASA Mars 2020 mission, to land the Perseverance rover, a computer vision system was used for hazards and obstacle detection to land safely by autonomously selecting the safest landing position [4]. Autonomous robots play an important role in various applications, where accurate perception and effective path planning are key requirements for achieving full autonomy. The perception component is dedicated to understanding the surrounding environment, enabling these robots to make informed decisions [5]. To have different fully autonomous vehicles in the future, accurate perception systems have become indispensable components, ensuring a reliable monitoring and interpretation of complex dynamic environments [6]. Moreover, the use of semantic segmentation techniques can exhibit higher precision in detecting urban environments [7]. Semantic data can help reduce the dependence of a robot on raw sensor input and external signals such as GPS by providing useful environmental information for navigation [8]. In fact, in this work the demonstrated approach for segmenting road scene images from a driver's perspective can also be used in other applications, such as delivery or taxi robots, and various service robots, that need to accurately understand their surroundings when navigating urban environments and avoiding obstacles. Among the most noticeable applications is parcel delivery, with autonomous delivery robots emerging as key components in the solution to different delivery challenges [9].

Autonomous robotic platforms operate in complex urban environments and require precise understanding and reliable path planning, while having limited processing resources. Many computer vision models often require extensive computational power, and this transformer-based system can offer a lightweight alternative, making it suitable for deployment across different autonomous robotic platforms. The ability of the autonomous robot to understand its working environment is the basis for solving more complicated problems [10]. The demonstrated results in this work will show that the proposed approach, which is effective in detecting objects at close distance, can be particularly advantageous to use with autonomous self-driving robots. These robots generally operate at lower speeds

compared to cars. While a car traveling at 60 km/h covers 100 meters in just a few seconds, a robot may take up to 30 seconds to travel the same distance. So, for the robot, distant objects in the environment are not as critical as they are for an autonomous car.

Autonomous systems and their effectiveness are highly dependent on their ability to navigate a complex and unstructured environment [11]. Recent advances have further improved robotic navigation by enabling real-time performance in critical tasks such as environment perception, obstacle detection, obstacle avoidance, path planning, and path tracking [12]. Within path planning, obstacle avoidance is a crucial task in robotics, as autonomous robot operation requires that they reach their destination without collisions [13]. Effective object detection strategies face static obstacles, such as infrastructure or parked vehicles, and dynamic obstacles, including pedestrians and moving vehicles, each presenting unique challenges to safe navigation. To prevent a potential collision with pedestrians, an accurate detection system enables autonomous robots to intervene early, reducing the risk of accidents [14]. So an accurate computer vision system is essential to ensure the safe and efficient operation of autonomous robots, especially in dynamic urban environments. The main solutions currently focus on understanding the environment through visual information using various computer vision techniques, machine learning, and algorithms [15]. The use of computer vision techniques enables robots to autonomously understand their surroundings, adapt their trajectory, and perform tasks such as maintenance or exploration without human intervention. In addition, for autonomous robots to navigate in urban environments, it is very important to navigate on designated paths, such as footpaths or sidewalks, and avoid areas such as grass to ensure both safety and social conformity. Robots deployed in public environments as autonomous delivery robots operate in spaces in which people live and work [16]. For example, package delivery robots must be able to identify and follow safe and appropriate routes that allow them to navigate autonomously in a manner that is not only efficient, but also socially acceptable to people sharing the environment [17]. Since robots increasingly share space with humans in everyday environments, ensuring safety is paramount [18]. Computer vision applications help improve the efficiency of transportation systems, increase their level of intelligence, and improve traffic safety [19]. Moreover, the integration of computer vision with robotics holds significant promise for environmental protection efforts by enabling more efficient resource management and reducing urban environmental impacts [20]. In general, the development of robotic vision is more than just a scientific curiosity or a passing trend. It marks a significant step forward in what machines can do and this can strongly impact our daily lives [21].

1.1. Evolution of Deep Learning Architectures

In 1989, French scientist Yann LeCun created one of the first convolutional neural networks, it was called LeNet-5. This neural network was designed for a handwritten digits recognition task. The emergence of the LeNet-5 architecture paved the way for the continued success of convolutional neural networks in performing high-complexity computer vision tasks, and encouraged researchers to explore the capabilities of convolutional neural networks in performing image segmentation tasks [22]. Among the different deep learning models, convolutional neural networks have achieved excellent performance in different computer vision tasks such as image classification, object detection, or digital image segmentation. Convolutional neural networks have become one of the most successful and widely used deep learning architectures in computer vision tasks over the past decade.

While for many years the convolutional neural network architectures was state of the art choice to make a computer vision tasks, situation changed in 2017, when the transformers architecture was introduced. Recently, vision transformers have emerged as a

competitive alternative to the long-standing convolutional neural networks for computer vision tasks. The transformer neural network was developed and introduced by the scientist Ashish Vaswani in 2017 [23]. These neural networks today compete with state-of-the-art convolutional neural network architectures in terms of efficiency and accuracy.

During the research, it was found that it is possible to create accurate computer vision models without using convolutional layers as main components. One such idea is to use the vision transformer neural network architecture for feature extraction, which applies an attention-based mechanism to input images and can achieve competitive efficiency in performing the computer vision semantic segmentation task [24]. The use of transformer-based architecture over traditional convolutional neural network as a main component for the feature extraction part is due to transformer architecture that improved the capability of global context understanding. The use of self-attention mechanisms in transformers architecture allows system to capture relationships in the image between distant regions. This approach enables the network to integrate information throughout the image. It is especially beneficial for semantic segmentation tasks that require full-scene understanding. Transformers processes the images as a sequence of patches, which gives the opportunity to model both global and local interactions without being constrained by convolutional fixed-size kernels. So, the results can be more robust, particularly in situations where the boundaries of the object and contextual cues are crucial [25].

Another important thing to use transformer architecture is robustness in variations. Attention mechanisms can dynamically focus on the most relevant parts of the image, making it more resilient to different changes like rotation, scaling, or occlusions. This adaptability in complex road scenes helps, where noise or local variations hinder performance. In addition, transformer architectures show improved performance as the data amount increases, making them highly scalable for large datasets [26].

1.2. Modern Approaches to Semantic Segmentation

Semantic segmentation is the core task and remains one of the most challenging tasks in computer vision, which involves the classification of every pixel in a digital image into a corresponding class. It gives complete context of the scene by incorporating the category of objects, the location and the shape of all elements of the scene, including the background [27]. However, it is more challenging and usually more time-consuming than object detection and requires more advanced techniques and more high-quality annotated training data [28]. Over the years, modern approaches have been introduced to perform semantic segmentation tasks. From early fully convolutional networks that started pixel-level predictions to advanced encoder-decoder architectures, which integrate multiscale feature fusion and context-aware processing [25,29]. Recently, the incorporation of attention mechanisms and transformer-based models opened new capabilities to capture long-range dependencies and global contextual information, pushing performance to new heights. The progression of modern techniques is shaping state-of-the-art performance in the computer vision field. Table 1 presents a systematic summary of these methods in the field of computer vision image segmentation, including their core architectures, benchmark datasets, and mIoU results.

Table 1. Comparison of different architectures

Method	Architecture	Dataset	Result (mIoU)
FCN	Fully Convolutional Network	PASCAL VOC	67.5
UNet	Encoder-Decoder	Cityscapes	83.6
DeepLabV3+	ASPP + Decoder	Cityscapes	79.6
SETR	Transformer	Cityscapes	76.7
Mask2Former	Hybrid CNN-Transformer	Cityscapes	83.3

Fully Convolutional Networks (FCN) were among the first breakthroughs in the semantic segmentation task. By replacing the fully connected layers of traditional CNNs with convolutional layers, FCNs enabled end-to-end pixel-wise prediction, which allowed models to generate spatially dense output by upsampling the low-resolution feature maps from the convolutional layers. The evaluation of FCN showed that this method could effectively segment images, establishing a solid baseline for subsequent models [29].

Building on the FCN framework, encoder–decoder architectures such as U-Net have become a popular approach to segmentation tasks. U-Net uses a symmetric architecture in which an encoder gradually reduces the spatial dimensions while capturing semantic features, and a decoder progressively upsamples the features to produce a prediction at the pixel level. Skip connections are used between the corresponding encoder and decoder layers to help recover spatial details lost during downsampling. The demonstrated U-Net approach is particularly effective for tasks that require precise detection of objects, such as segmentation of the road scene [30].

A DeepLabV3+ model uses dilated convolution-based approach, to better capture multiscale contextual information without losing resolution. Using this type of approach increases the receptive field of convolutional filters, without additional parameters or loss of spatial resolution. Using spatial pyramid pooling and encoder-decoder structure, the DeepLabV3 + model can fuse features from multiple scales, which is critical for the complex and varied environments encountered in road scenes [31].

Adaptation of transformer architectures is another trend. Although originally developed for natural language processing, it was later used in the domain of image segmentation. Models such as SETR or SegFormer use the self-attention mechanism to capture long-range dependencies and global context. The images in these models are partitioned into patches and then processed as token sequences, enabling the network to model relationships between distant regions. The global modeling capability is especially advantageous for complex scenes, such as urban road environments, where contextual signals are critical [32,33].

The integration of convolutional neural networks with transformer modules showed a promising direction, combining the strengths of both architectures. Hybrid computer vision models often use convolutional layers for efficient local feature extraction and are combined with additional transformer layers to capture global context details through self-attention mechanisms. The use of this approach can lead to improved segmentation performance, especially in scenarios where both detailed spatial information and a wider contextual understanding are necessary. The work incorporating the convolution technique into visual transformers shows that using those two methodologies can yield competitive performance in segmentation tasks [34].

The main purpose of this study is to develop a transformer-based semantic segmentation approach, specifically designed for the road scene segmentation task, which takes advantage of the latest advances in vision transformers. The paper is further organized as follows. Section 2 describes our proposed semantic segmentation approach. Section 3

presents the experimental results, and Sections 4 and 5 conclude the whole document with a discussion and future directions.

2. Materials and Methods

In this section, we provide all the details about the implementation, structure of the model, and configuration settings used to train and evaluate our transformer-based road scene segmentation system. Also, we describe what other state-of-the-art computer vision models mainly use, what we used on our approach, and how this approach is different, and some advantages over the previous used implementations. In addition, we describe the dataset in more detail and how the data are divided into training, validation, and test set splits. In further sections we also explain in details about internal structure of the encoder module, and our approach with using a decoder part of the system with attention based fusion mechanisms for extracting and combining multiscale features. More details of what was used in each component structure will be provided in further subsections.

Current state of the art computer vision delivered approaches for road scene segmentation demonstrated significant success by using different types of architectures and reaching competitive results. Methods that use transformer-based architecture typically rely on standard feature fusion behavior, and often they use simple projection layers to merge different features from different transformer-based encoder resolutions. Computer vision models also introduced, such as RoadFormer, use a query-based decoder module to iteratively update query features for mask predictions [35]. Although these demonstrated approaches work well in many cases, sometimes they can struggle to maintain important fine-grained details when working with complex road scene environments.

Our decoder module approach processes the data with attention-based fusion mechanisms, providing competitive results on the road scene segmentation task with complex urban scenes compared to other lightweight computer vision models that have a similar amount of parameters in the network. The mechanism in the decoder part uses specialized attention blocks to progressively fuse low-resolution decoder features with high-resolution features from the encoder part. This can effectively emphasize the most relevant spatial information on multiple scales. Using this approach, it is possible to ensure that fine-grained features of the encoder can be integrated together with high-level semantic features of the decoder, giving accurate and competitive segmentation results when compared with other introduced approaches. Also, the model instead of using traditional normalization techniques uses group normalization, which normalizes over the feature channels instead of making that on batch dimension. This can give better stability in the training process, particularly in such scenarios where the batch size may be different, or in cases of high-resolution feature fusion, where traditional methods such as BatchNorm may not be as effective.

Also worth mentioning that while many existing approaches rely on standard different scales feature fusion techniques, this approach can provide better control over which features are fused, by use of attention blocks, which allow to adapt more effectively to the different complexities of road scene segmentation. Using this approach gives competitive results on segmentation accuracy, particularly in challenging road scene environments with fine details.

2.1. Dataset

All experiments were made using the widely known Cityscapes dataset, which is used for benchmarking a computer vision models on semantic segmentation task with different urban scene images. The data set consists of 5000 high-quality finely annotated pixel level high-resolution images recorded in 50 different cities across Germany. Each pixel in every

image in the data set is annotated in one of 19 semantic categories, representing typical elements of the urban road scene such as the road, car, pedestrian, sidewalk, or traffic signs. The part of the data set of 5000 images is split into:

- **Training set consists of 2,975 images:** This subset of data is used to learn the parameters of the model during the training process.
- **Validation set consists of 500 images:** This part of the data is used during the training process to configure the model, adjust hyperparameters and monitor the performance of the model during the training phase.
- **Test set consists of 1,525 images:** The last part of the data is used for the final evaluation phase to evaluate the precision of the model in an unseen data set.

Figure 1. Sample images from the Cityscapes dataset, illustrating the diversity of urban scenes in different cities across Germany.

In figure 1 we can see a couple of examples from the Cityscapes data set, demonstrating different environment of urban conditions. All details about the data set used in the model training process and the final evaluation, also about annotated classes, and examples of annotations are publicly available at https://www.cityscapes-dataset.com/ with the full class list and additional details about other subsets.

2.2. Implementation Details

Our computer vision transformer-based road scene segmentation model is implemented using the `mmsegmentation` framework codebase and uses a MiT as a backbone network with attention-based fusing mechanisms in the decoder module. The systems encoder module is pre-trained on the ImageNet-1k dataset for extracting robust visual features, while the system's decoder module is randomly initialized to learn task-specific upsampling. To improve model robustness and generalization, in the training process, data augmentation techniques were applied just using a Cityscapes dataset. We used additional functionality of random horizontal flipping, scaling, and cropping. The crop size of 768×768 pixels was chosen during the training phase. At inference time, a sliding-window strategy was used to generate full-size segmentation predictions.

The computer vision model was trained using an AdamW optimization algorithm with an initial learning rate preset to 0.00005. In addition, the polynomial decay learning rate schedule with the power parameter set to 1.0 was used, and the linear warm-up phase was used up to 1500 iterations at the start of the training process. Due to GPU resource

constraints and high-resolution input images, a batch size of 1 image was used in the training process. The training schedule was set for 160 000 iterations, and additionally later on was extended to 200 000 iterations to explore the possibility of further model's accuracy improvements. In the end, it was clear that no significant improvements were achieved in the training cycle above 160 000 iterations. The best performing model checkpoint was taken at 156 000 iterations. Model performance was primarily evaluated using the widely used mean intersection-over-union (mIoU) accuracy metric.

2.3. Encoder

In the system, the encoder backbone network is implemented using the MiT architecture, with configuration settings named "mit b1" [33]. Using these configuration settings, the input image is processed through a patch embedding module that divides the image into smaller patches and projects them into a feature space. The first stage of the network uses an OverlapPatchEmbed module with a fixed patch size of 7x7 and a stride parameter of 4. The module is responsible for mapping each 7x7 image patch to 64-dimensional embedding space and generating the initial feature map. The patch embedding can be written down as:

$$X_0 = \text{Conv}_{7\times7}(I)$$

where I is the input image. The output after this processing is then flattened and normalized. The subsequent stages of the encoder network use patch embedding modules with a set patch size of 3x3 and a stride parameter of 2. These network encoder stages progressively reduce the spatial resolution, while increasing the number of channels. Using the "mit b1" configuration, the embedding dimensions are preset to [64, 128, 320, 512] for the four network stages. The encoder progressively extracts hierarchical features with increasing channel numbers:

- Stage 1 (c1): 64 channels;
- Stage 2 (c2): 128 channels;
- Stage 3 (c3): 320 channels;
- Stage 4 (c4): 512 channels.

Each of the stages is further processed by a sequence of transformer blocks. Using the b1 network configuration, the depth of each block is [2, 2, 2, 2] for the four stages. In addition, with each transformer block, the multi-head self-attention mechanism is used. The attention computation can be written as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

where Q is a query, K is the key, and V is the value matrices received by linear projections of the input features. In each stage, the number of attention heads used is [1, 2, 5, 8], and the MLP within each block expands the dimension of the feature by a factor of 4.

In each stage, spatial reduction ratios are also used with values [8, 4, 2, 1] to downsample the spatial dimensions during the self-attention processing step. By doing this, it reduces the computational load while maintaining an essential global context.

After the processing step is performed through different transformer blocks, each stage uses a final normalization layer, and the resulting feature maps are reshaped in the [B, C, H, W] format. The final outputs of all four different stages are named c1, c2, c3, and c4. These outputs capture a rich hierarchy of features that combine fine spatial details from the early stages of the network with high-level semantic information from the deeper stages of

the network. In a further step, all these generated feature maps from each encoder network stage are passed to the decoder module for further processing.

2.4. Decoder

The following decoder module approach combines features from the encoder using an attention-based mechanism. In this subsection, we propose an Attention-Guided decoding module that fuses multiscale transformer features via hierarchical attention and residual convolutional fusion. The decoder design lets the rich semantic information from deeper layers be combined with the fine spatial details of shallower layers, and by doing so it can achieve good accuracy and spatial precision. Figure 2 shows the complete model architecture, including the encoder and the Attention-Guided decoding module.

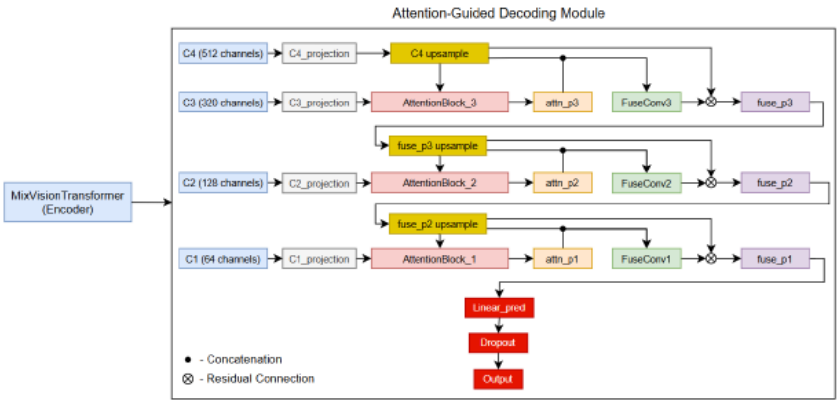


Figure 2. Architecture diagram of a computer vision model with the proposed Attention-Guided decoding module.

Using MiT as a backbone feature extraction network, the encoder part produces a set of feature maps at different resolutions, typically named c1, c2, c3, and c4 from the shallowest to the deepest network stage. Each feature map captures information on a different scale. Although deeper features contain higher semantic information, shallow features preserve detailed spatial information, which is critical for accurate segmentation in complex road scenes.

The decoder first projects each of the feature maps in a common embedding space with a fixed number of channels. This process is achieved by using a 1x1 pixel convolutional layer, which is later followed by normalization of feature maps across channels to help improve the consistency of feature representations. After that, the ReLU activation function is also used to introduce non-linearity, by letting the network to learn more complex patterns. This mentioned process not only standardizes the channel dimensions across different scales but also improves the stability of the whole training process. Mathematically, this projection can be described as follows:

$$p_i = \mathrm{ReLU}\Big(\mathrm{GroupNorm}\big(\mathrm{Conv}_{1\times1}(c_i)\big)\Big), \quad i \in \{1,2,3,4\}.$$

In addition, in the decoder part, it uses an attention-based fusion mechanism. When using separate 1x1 convolutional and normalization layers for upsampled decoder features and corresponding encoder features, it generates an attention map that highlights the most relevant spatial regions. This selective weighting approach helps the decoder focus on important details. As a result, the integration of rich semantic features with fine spatial details helps to achieve a more effective and accurate segmentation process in different

road scenes. The attention block in the decoder part is a main component that serves to fuse features from different scales. In this decoder implementation, the fusion process is performed in a top-down approach. In the first stage, the deepest feature map p4 is first upsampled to match the spatial resolution of p3. After that, the attention block receives the upsampled p4 feature map (acting as a gating signal) and the p3 feature map from the encoder backbone. Later, the attention block projects received both inputs using a 1×1 convolutional layer followed by a normalization layer. Specifically, the projections can be formulated as follows:

$$g_1 = \text{GroupNorm}\Big(\text{Conv}_{1\times1}(g)\Big), \quad x_1 = \text{GroupNorm}\Big(\text{Conv}_{1\times1}(x)\Big),$$

where g is the gating signal(upsampled p4) and x is the encoder feature (p3). The resulting outputs are then summed together, activated by the ReLU function, and further processed by another 1×1 convolutional layer with the sigmoid activation function to produce an attention map:

$$\psi = \sigma\Big(\text{GroupNorm}\big(\text{Conv}_{1\times1}\big(\text{ReLU}(g_1 + x_1)\big)\big)\Big).$$

This produced map selectively weights the p3 features, suppressing less relevant regions. The weighted encoder feature is concatenated with the upsampled p4, and a convolutional module refines this fusion. Additionally, a residual connection is added by adding the upsampled p4.

In addition, the fused features from the previous stage are up-sampled in the same way as the resolution of p2 and fused with p2 using an analogous attention-guided procedure. The same process is repeated once again with the highest-resolution p1 feature map to produce the final fused representation.

In this decoder approach, there are combined residual connections at each fusion stage. These connections add the upsampled features back into the fused output, and by doing this, it ensures that essential spatial details are kept throughout the upsampling process. By using residual connections at each stage, it helps in the training process to stabilize it by maintaining a smoother gradient flow. In addition, it allows the model to take advantage of the refined fused features and the original upsampled signals, thereby improving the overall robustness of the segmentation process.

After the progressive fusion step, a dropout layer is applied to the refined feature map for regularization. After this step, the final 1×1 convolutional layer is used to project the features to the preset number of segmentation classes, generating the final segmentation output. The final prediction can be written as:

$$\text{Output} = \text{Conv}_{1\times1}\Big(\text{Dropout}(\text{Fused Feature})\Big)$$

This demonstrated decoder part approach can be effective for road scene segmentation tasks due to its ability to preserve spatial precision by maintaining high-resolution feature maps and integrating fine-grained details using the attention mechanism, which ensures accurate distinction of road boundaries and different objects. The attention mechanism used in the decoder module works as a gating function which merges context-rich deep features with detailed shallow features. In doing so, it emphasizes the spatial regions that are most relevant. In addition, the use of residual connections throughout the fusion process helps with training stability and convergence, even when the model incorporates multiple upsampling stages.

Overall, this decoder approach provides a robust and efficient capability for fusing multiscale features, and when combined with the MiT backbone for feature extraction, it can achieve competitive good accuracy in the road scene semantic segmentation task.

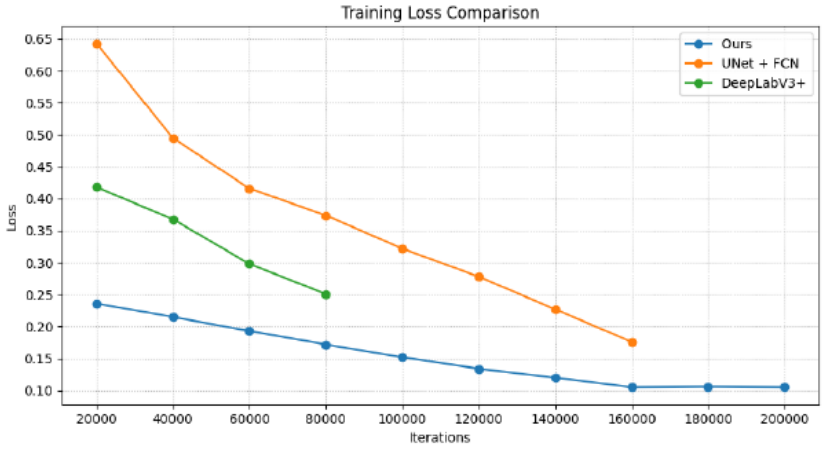


Figure 3. Training loss curves of our model versus U-Net and DeepLabV3+, showing faster convergence and lower final loss.

The results show the efficiency of our proposed training approach. Although we extended training to 200 000 iterations, we saw no further improvement beyond the 160 000 iterations line, so continuing past that point would not be cost-effective.

3.2. *Per-Class Performance*

Our computer vision model segments urban road scenes into 19 different classes according to the standard of the Cityscapes data set. It uses the cross-entropy loss function for optimization. For better understanding, we divide the classes into two different groups based on the segmentation accuracy they reached, measured by the main intersection over the union accuracy metric values. The first group of classes consists of reaching the highest mIoU accuracy, which is above $\geq 76\%$, while the second group of classes indicates those who reach the mean intersection over union accuracy far below the 76% line. It is worth mentioning that despite training exclusively only on Cityscapes dataset, and without using any additional datasets, the computer vision model is capable of reaching a pretty good accuracy on semantic segmentation task while having less than 20 million parameters count and detecting different classes such as road, cars, sidewalks, vegetation, or buildings.

In Table 2 we can see the first group of classes that reached the higher accuracy than 76%, ordered in descending order by the IoU metric for every class.

Table 2. Classes which have higher accuracy than 76% in descending order.

Class	IoU (%)	Accuracy (%)
Road	97.93	98.83
Sky	94.54	98.15
Car	94.46	98.01
Vegetation	92.43	96.84
Building	92.16	96.43
Sidewalk	83.39	91.63
Person	80.90	91.60
Bus	80.04	85.36
Truck	79.17	84.65
Traffic Sign	78.83	85.16
Bicycle	76.71	87.72

The table above demonstrates that classes, such as road, car, sky, vegetation and building, are reaching the accuracy above 90%. Others, like a bus, person, sidewalk, truck, traffic sign, and bicycle, reach lower accuracy metrics, but still not bad, while keeping in mind the lightweight model approach, the amount of data used, and how difficult sometimes it can be to distinguish classes like traffic signs in different distance conditions. In the following, Table 3 lists the classes with mIoU below 76%, indicating that they are more challenging to accurately segment.

Table 3. Classes which reach less accuracy than 76% and more difficult to detect.

Class	IoU (%)	Accuracy (%)
Traffic Light	70.19	81.64
Motorcycle	65.86	76.66
Train	64.97	69.36
Terrain	63.02	69.89
Pole	62.83	72.81
Fence	59.84	70.10
Rider	59.45	71.76
Wall	55.14	62.86

Table 3 indicates that classes such as motorcycle, train, traffic light, pole, terrain, fence, wall, and rider reach lower precision metrics. These challenges may be likely due to the inherent complexity and variability of these objects in different road scenes. In the future, by using more data with bigger diversity, it is possible to get better accuracy results, and for these classes, too. Here remains the challenge of detecting classes more accurately, such as the wall or fence, because it is a challenging task when a distinction is needed between a concrete wall, a building, or a fence for the model, and many state-of-the-art models lack the accuracy of these classes. Overall, the computer vision model, considering that it has 17.2 million parameters, detects different main objects and the environment such as cars, roads, and sidewalks with good accuracy, while the lower performance in certain classes highlights opportunities for further future improvements.

3.3. Global Metrics

To evaluate the final performance of our computer vision model with the Cityscapes validation set of data, we collected different accuracy metrics. The following accuracy metrics were obtained after the final evaluation phase:

- **Mean IoU (mIoU):** 76.41%
- **Mean Accuracy (mAcc):** 83.66%
- **Overall Accuracy (aAcc):** 95.87%

The above provided metrics demonstrate that the computer vision model, having 17.2 million parameters count, is capable of reaching global 76.41% of mean intersection over the union accuracy metric. This metric reflects the average overlap between the predicted segments and the ground truth values across all classes, and it is mostly used for evaluating the performance of the most computer vision models. The data provided above also show that the model reached the mean accuracy (mAcc) value of 83.66%. This metric represents the average classification accuracy per pixel for each class. Lastly, the overall accuracy metric (aAcc) of 95.87% is reached. This metric shows the ratio of correctly classified pixels to the total number of pixels in the Cityscapes validation part set.

3.4. Qualitative Analysis

In Figure 6 we can see an example image that shows a sample road scene where the original image on the left is presented side-by-side with the segmentation output produced by the system on the right side. From this side-by-side comparison, it is clearly visible how the model can perform a semantic segmentation task on the road scene images detecting different objects in the digital image with good accuracy. When making a comparison of two images side-by-side, we can state that:

- **Accurate detection of close objects:** The model is capable of detecting near-field classes with pretty good accuracy like roads, cars, pathways, buses when these objects are approximately in 50-100 meters distance.
- **Challenges with distant objects:** It is more challenging to detect distant objects like traffic signs, indicating potential areas for further improvement.
- **Overall performance:** Under ideal weather conditions, segmentation quality is pretty good, the model clearly defines boundaries for different objects, although minor inaccuracies can appear in more complex or distant regions.

Figure 4. Road scene example with the original image on the left, and the segmented one on the right.

The segmentation result provided above demonstrates that the computer vision model developed can firmly detect and segment close objects, ensuring clear and detailed recognition of the main elements on the road. It should be mentioned that lower accuracy metrics were reached for more distant objects such as traffic signs, poles, or traffic lights. This can leave a possible area for improvement in the future. The main focus can be improving the ability of the computer vision model to detect fine details in more distant regions. It can be achieved by modifying the model architecture or by using additional data sets. By doing that, it is possible to further enhance the performance of the system in different real-world scenes on the road.

3.5. Model Robustness Analysis

To evaluate the robustness of our segmentation model, we also tested it on images of a different data set (Mapillary Vistas) that were not used during training in any form or shape. The examples below include a couple of images captured in an urban city area, which highlights the model's performance in a different environment, and as an example another image taken on a highway to demonstrate model capability to accurately segment unseen data across different scenarios.

Figure 5. Segmentation results with unseen images from another data set.

In Figure 5, on the left is the original image and on the right is the corresponding segmented result. The visual presentation demonstrates the ability of the model to detect major objects and environment with good overall accuracy. It also indicates that there is room for improvement in the segmentation of small objects, such as traffic signs or traffic lights, and that the accuracy at larger distances could be improved. For example, in the highway photo, where the distance is more than 100 meters, the model lacks the accuracy to differentiate cars from the road, but with closer objects and environment, the accuracy is good.

Table 4. Class Performance Metrics

Class	IoU (%)	Accuracy (%)
Road	96.72	98.29
Sky	93.05	96.60
Car	90.58	96.42
Vegetation	89.44	96.43
Building	89.18	94.64
Sidewalk	75.57	84.58
Person	67.59	83.94
Traffic Sign	62.63	69.76

As demonstrated in Table 4, the highest performance metrics have the main environment and object classes such as roads, cars, buildings, vegetation, which indicates a good segmentation performance for these dominant regions. In addition, classes such as person or traffic signs exhibit lower performance metrics, which can be attributed to their smaller size and higher variability in appearance.

3.6. Computational Performance Analysis

The computer vision model, with 17.2 million parameters, was evaluated on a NVIDIA GTX 1060 GPU, chosen specifically to represent the low end sector of Cuda capable devices (released in 2016). The test was performed using a single image with a batch size of 1, with different input resolutions. The computational performance metrics such as computational

cost (GFLOPs), inference time (ms), and throughput (FPS) at different resolutions are provided in the table 5, illustrating the model's resource requirements under the different conditions. As resolution increases, both computational cost (GFLOPs) and inference time increase, while throughput (FPS) decreases. Specifically, at 256×256 resolution, the model requires 9.1 GFLOP and achieves 42.4 FPS with an inference time of 24 ms. With the highest resolution of 2048×1024, the computational cost reaches 419.4 GFLOPs, and the inference time increases to 769 ms. The data provided in the table demonstrate the trade-offs between resolution, computation, and performance that must be carefully considered when a different resolution is needed for resource-constrained applications.

Table 5. Computational Performance at Different Input Resolutions

Input Resolution	GFLOPs	Inference Time (ms)	Throughput (FPS)
256×256	9.1	24	42.4
512×512	37.9	37	26.9
1024×1024	176.4	120	8.3
2048×1024	419.4	769	1.3

In Table 6, we present the computational performance of our segmentation model on the NVIDIA Jetson Orin Nano 8GB embedded platform. Performance metrics are reported in three different input resolutions, capturing inference speed (FPS) and GPU memory usage.

Table 6. Model performance on Jetson Orin Nano 8GB

Input Size	Inference Speed (FPS)	GPU Memory Usage (MiB)
256×256	52.89	54
320×320	32.23	72
512×512	15.56	112

The results provided in Table 6 demonstrate that at the lowest resolution, the computer vision model can achieve an inference speed of 52.89 FPS while consuming 54 MiB of GPU memory. When using a higher resolution, the inference speed decreases to 32.23 FPS at an input size of 320×320. At 512×512 resolution, the GPU memory consumption increases to 112 MiB and 15.56 FPS is still acceptable for many autonomous applications. However, for applications that require higher input resolutions or faster inference speeds at larger scales, a more powerful embedded device with greater computational resources may be more appropriate than the Jetson Nano embedded platform.

4. Discussion

The experimental results of the computer vision road scene segmentation system demonstrated pretty good accuracy and effectiveness when the model uses a transformer-based MiT backbone as an encoder for feature extraction and attention-based mechanisms in the decoder module. During the training log analysis, the results revealed that the model, during the whole training cycle, was steadily reducing its loss parameter, and just before reaching the final iterations, it demonstrated the best accuracy performance at the 156,000 iterations checkpoint. Furthermore, the training process confirmed that setting the training schedule at 160,000 iterations was cost efficient and effective because an additional experiment extending the training process was no longer effective. In addition, the per-class performance metrics show that the main elements of the road scenes, such as the road, cars, buildings, are segmented with pretty good accuracy, indicating the model's good

performance capability to detect and distinguish different objects and environment on the road.

Creating a model that maintains a lightweight design with a low parameter count while achieving competitive accuracy metrics is a tricky task. It is always necessary to have a balance between accuracy and lightweight model design, because while a more complex model can reach better accuracy metrics, it also has a significantly larger number of parameters and is consequently not as lightweight. This balance is critical, as increasing model complexity often improves performance, but can compromise the efficiency and suitability of the model for deployment in memory-limited environments.

According to global evaluation metrics of the developed model, they validate the good performance of the system capacity to detect and differentiate objects in digital road scene images, achieving a mean IoU 76.41% metric with 17.2 million parameters in the network. Despite its compact size, the computer vision system is capable of maintaining competitive segmentation performance results, making it suitable for use in memory-limited resource environments. The visual examples provided in Figure 6 show that the model is pretty good at segmenting near-field objects within 50-100 meters distance, and classes like roads, cars, and pathways are detected with good accuracy. However, it is worth mentioning that more distant objects such as poles, traffic lights, traffic signs, also the fence class has less precision with classes mentioned in the first group with higher accuracy metrics. So, these lower-metric classes can be an avenue for future system enhancement by incorporating additional training data or modifying the internal system architecture to capture small-scale objects or environments in distant regions. In Figure 6, we provide qualitative results with the original RGB image on the left, the ground truth masks in the middle, and the segmentation results on the right.

Figure 6. Qualitative segmentation results. The left column shows the original RGB image, the middle column shows the ground truth masks, and the right column shows the model's segmentation results.

According to other open-source computer vision proposed approaches, models such as DSNet [36], HRNetV2 + OCR [37], DeepLabv3 + [37], CSFNet-1 [38], and EEEA-Net-C2 [39], have demonstrated good accuracy results on the Cityscapes dataset while maintaining relatively low parameter count. And this demonstrated approach has 17.2 million parameters and reaches 76.41% mIoU metric. This approach also demonstrates that by using a transformer-based encoder for feature extraction and an attention-based mechanism to fuse

multi-scale features, it is possible to achieve competitive accuracy with other models while maintaining a lightweight design and good overall performance. This developed approach can also add more knowledge to existing studies that transfomer-based architectures can effectively segment complex urban scenes with models that have a low parameter count and are capable of reaching good accuracy in road scene semantic segmentation tasks. Table 7 shows details of several semantic segmentation models evaluated in the Cityscapes validation set, including their input resolutions, parameter counts, GFlops, and mIoU values as reported in the literature.

Table 7. Comparison of Semantic Segmentation Models

Model	Input Size	Params (M)	GFLOPs	mIoU (%)
EEEA-Net-C2	512×512	7.34	28.7	76.8
CSFNet-1	1024×512	12.6	86.9	74.8
Ours	512×512	17.2	37.9	76.4
DSNet	2048×1024	37.5	226.6	82.0
DeepLabv3+	2048×1024	43.5	1444.6	79.6
HRNetV 2 + OCR	2048×1024	70.3	1206.3	81.6

The computer vision models in Table 7 are sorted by the number of parameters in the network. From the data provided in the table, we can see that our approach has a balance between segmentation accuracy, model size, and computational efficiency. The proposed model has 17.2 million parameters and requires 37.9 GFlops at an input resolution of 512x512. Models like EEEA-Net-C2 and CSFNet-1 represent lightweight architectures, achieving mIoU accuracy metrics of 76.8% and 74.8%, respectively. In comparison, a model like CSFNet-1 has a slightly lower parameter count but a larger computational cost. A model like DSNet gives good accuracy results and computational efficiency, but it has more than twice as many parameters in the network. Also, models like DeepLabV3 + and HRNetV 2 + OCR reach better accuracy results but at the expense of much larger architectures. Models like HRNetV2 + OCR achieve higher segmentation accuracy, but their architecture has roughly four times as many parameters and requires significantly more computational resources compared to our approach. This comparison highlights that the proposed approach offers a compelling trade-off by delivering good segmentation performance in a lightweight architecture.

In Table 8, we compare the Intersection-over-Union (IoU) per class of our lightweight segmentation model against two much larger architectures: HRNet-W48 (65.9 M) and DeepLabV3-R101 (84.7 M). Despite having roughly one-quarter to one-fifth the number of parameters, our approach achieves only marginally lower IoU scores across many semantic categories. The side-by-side comparison makes it clear that although larger architectures can reach out slightly higher accuracy metrics, our transformer-based design with attention-guided decoding module offers a compelling trade-off, maintaining competitive performance while drastically reducing model complexity.

Table 8. Per-class segmentation accuracy (IoU) compared across different network architectures.

Class	DeepLabV3-R101 (84.7 M)	HRNet-W48 (65.9 M)	Our (17.2 M)
Road	98.3	98.4	97.9
Car	95.4	95.6	94.5
Sky	94.2	95.2	94.5
Building	92.5	93.4	92.2
Vegetation	92.0	93.0	92.4
Sidewalk	85.7	86.4	83.4
Person	81.3	83.4	80.9
Traffic Sign	78.7	81.7	78.8
Bicycle	77.8	79.0	76.7
Traffic Light	69.8	72.4	70.2
Motorcycle	69.2	67.1	65.9
Terrain	63.3	66.7	63.0
Rider	63.9	62.3	59.5
Pole	58.9	69.3	62.8
Fence	62.7	66.4	59.8
Wall	53.5	59.7	55.1

In addition to the overall quantitative comparison, Table 8 shows that our transformer-based encoder with the attention-guided decoding module matches the larger HRNet-W48 and DeepLabV3-R101 architectures almost exactly in large and homogeneous classes such as road, car, building, and vegetation, with IoU differences of less than 1.2 percentage points. The results show that the global context modeling of the MiT backbone effectively captures broad, texture-rich regions. However, all three architectures, including the larger baselines, have the greatest difficulty accurately segmenting slender or distant objects, such as poles, fences, and walls.

In summary, this demonstrated approach confirms that a computer vision system based on the transformer neural network architecture with an attention-guided decoding module can effectively capture contextual information for the road scene segmentation task. While the model performs pretty well with near-field objects, future work can put more attention on areas detecting more distant small-scale objects, because many models are lacking this type of accuracy and it still remains the challenging task.

5. Conclusions

In this paper, we introduced an approach that uses a transformer-based neural network architecture to segment road scene images, using the MiT backbone as a feature extractor and an attention-guided decoder to effectively fuse multiscale features. Experimental results with the Cityscapes dataset revealed that, despite the compact parameter count, it can achieve a competitive mean IoU 76.41% result, in different semantic segmentation tasks of urban scenes. The conducted training process confirmed that the system is able to converge reliably in 160 000 iterations, and the additional training process with the same configuration and data set did not produce better accuracy results.

The model evaluation phase and visual data highlighted that the main objects in close proximity can be detected with good accuracy, while the lower accuracy classes can be an avenue for future accuracy improvement. In general, the primary design goal of this approach was to develop a lightweight computer-vision road scene segmentation model. This transformer-based segmentation model approach has only 17.2 million parameters, which is remarkably low compared to many computer vision semantic segmentation models. The demonstrated model can ensure a small memory footprint, making it well suited for deployment on memory-limited devices.

Author Contributions: Conceptualization, Rytis Maskeliunas; Formal analysis, Bartas Lisauskas and Rytis Maskeliunas; Funding acquisition, Rytis Maskeliunas; Investigation, Bartas Lisauskas and Rytis Maskeliunas; Methodology, Bartas Lisauskas; Project administration, Rytis Maskeliunas; Resources, Bartas Lisauskas; Software, Bartas Lisauskas; Supervision, Rytis Maskeliunas; Validation, Bartas Lisauskas; Visualization, Bartas Lisauskas; Writing – original draft, Bartas Lisauskas; Writing – review & editing, Rytis Maskeliunas.

Data Availability Statement: All data is freely available on the indicated datasets.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
FLOPS	Floating Point Operations per Second
FPS	Frames per Second
GPS	Global Positioning System
IoU	Intersection over Union
mAcc	Mean Accuracy
mIoU	mean Intersection over Union
MLP	Multilayer Perceptron
MiT	Mix Transformer
NASA	National Aeronautics and Space Administration

References

1. Zohaib Rafique. Improving Efficiency of Computer Vision for Autonomous Vehicles, 2020. https://doi.org/10.13140/RG.2.2.11761.92009.
2. Fernandes, S.; Duseja, D.; Muthalagu, R. Application of Image Processing Techniques for Autonomous Cars. *Proceedings of Engineering and Technology Innovation* **2020**. https://doi.org/10.46604/peti.2021.6074.
3. Sellat, Q.; Bisoy, S.; Priyadarshini, R.; Vidyarthi, A.; Kautish, S.; Barik, R.K. Intelligent Semantic Segmentation for Self-Driving Vehicles Using Deep Learning. *Computational Intelligence and Neuroscience* **2022**, *2022*, 1–10. https://doi.org/10.1155/2022/6390260.
4. NASA. Mars 2020 Perseverance Rover, 2020. Accessed: 2025-03-19.
5. Chen, W.; Chi, W.; Ji, S.; Ye, H.; Liu, J.; Jia, Y.; Yu, J.; Cheng, J. A Survey of Autonomous Robots and Multi-Robot Navigation: Perception, Planning and Collaboration. *Review* **2025**.
6. Iparraguirre-Gil, O. Computer Vision and Deep Learning based Road Monitoring towards a Connected, Cooperative and Automated Mobility **2025**.
7. Marasinghe, R.; Yigitcanlar, T.; Mayere, S.; Washington, T.; Limb, M. Computer Vision Applications for Urban Planning: A Systematic Review of Opportunities and Constraints **2024**.
8. Landsiedel, C.W. Semantic Mapping for Autonomous Robots in Urban Environments. PhD thesis, Technische Universität München, 2018. Doctoral dissertation.
9. Boysen, N.; Fedtke, S.; Schwerdfeger, S. Last-mile Delivery Concepts: A Survey from an Operational Research Perspective **2020**.
10. Ni, J.; Chen, Y.; Shi, P.; et al. Deep Learning-Based Scene Understanding for Autonomous Robots: A Survey **2023**.
11. Ogunsina, M.; Efunniyi, C.P.; Osundare, O.S.; Folorunsho, S.O.; Akwawa, L.A. Reinforcement Learning in Autonomous Navigation: Overcoming Challenges in Dynamic and Unstructured Environments **2024**.
12. Adiuku, N.; Avdelidis, N.P.; Tang, G.; Plastropoulos, A. Advancements in Learning-Based Navigation Systems for Robotic Applications in MRO Hangar: Review. *Review* **2025**.
13. Katona, K.; Neamah, H.A.; Korondi, P. Obstacle Avoidance and Path Planning Methods for Autonomous Navigation of Mobile Robot **2024**.
14. Janai, J.; Güney, F.; Behl, A.; Geiger, A. Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art **2021**.

15. Conde, M.V. An Embarrassingly Pragmatic Introduction to Vision-based Autonomous Robots: Applications, Datasets and State of the Art. *Review* **2021**.
16. Pelikan, H.R.M.; Reeves, S.; Cantarutti, M. Encountering Autonomous Robots on Public Streets **2024**.
17. Buckeridge, S.; Carreno-Medrano, P.; Cosgun, A.; Croft, E.; Chan, W.P. Autonomous Social Robot Navigation in Unknown Urban Environments Using Semantic Segmentation. *Review* **2021**.
18. Siegwart, R.; Nourbakhsh, I.; Scaramuzza, L. *Introduction to Autonomous Mobile Robots*, second edition ed.; MIT Press, 2011.
19. Dilek, E.; Dener, M. Computer Vision Applications in Intelligent Transportation Systems: A Survey **2023**.
20. Che, C.; Zheng, H.; Huang, Z.; Jiang, W.; Liu, B. Intelligent Robotic Control System Based on Computer Vision Technology **2024**.
21. Upadhyay, D.; Upadhay, D.K.; Singh, R.; Mishra, D.; Khatri, V. Robotic Vision: Advancements in Computer Vision for Autonomous Systems **2023**.
22. Sultana, F.; Sufian, A.; Dutta, P. Evolution of Image Segmentation using Deep Convolutional Neural Network: A Survey. *Knowledge-Based Systems* **2020**, *201–202*, 106062. https://doi.org/10.1016/j.knosys.2020.106062.
23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need, 2023, [arXiv:cs.CL/1706.03762].
24. Bai, Y.; Mei, J.; Yuille, A.; Xie, C. Are Transformers More Robust Than CNNs?, 2021, [arXiv:cs.CV/2111.05464].
25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is All You Need. In Proceedings of the Advances in Neural Information Processing Systems, 2017.
26. Zhai, X.; Kolesnikov, A.; Houlsby, N.; Beyer, L. Scaling Vision Transformers. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.
27. Hurtado, J.V.; Valada, A. Semantic Scene Segmentation for Robotics, 2024, [arXiv:cs.RO/2401.07589].
28. Wang, Y.; Ahsan, U.; Li, H.; Hagen, M. A Comprehensive Review of Modern Object Segmentation Approaches. *arXiv* **2023**.
29. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *arXiv preprint arXiv:1411.4038* **2014**.
30. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer, 2015, pp. 234–241.
31. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 801–818.
32. Zheng, Y.; Tan, X.; Zheng, Z.; Zhou, Y.; Li, B.; Yi, L. Rethinking Semantic Segmentation with Transformers. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021.
33. Xie, E.; Wang, W.; Yu, Z.; An, T.; Gao, Y.; Lu, M.; Xu, X.; Ren, T.; Zhang, C.; Xiao, T.; et al. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. *arXiv preprint arXiv:2105.15203* **2021**.
34. Yuan, K.; Guo, S.; Liu, Z.; Zhou, A.; Yu, F.; Wu, W. Incorporating Convolution Designs into Visual Transformers. *arXiv preprint arXiv:2103.11816* **2021**.
35. Li, J.; Zhang, Y.; Yun, P.; Zhou, G.; Chen, Q.; Fan, R. RoadFormer: Duplex Transformer for RGB-Normal Semantic Road Scene Parsing. *IEEE Transactions on Intelligent Vehicles* **2024**, *9*, 5163–5172. https://doi.org/10.1109/tiv.2024.3388726.
36. Guo, Z.; Bian, L.; Huang, X.; Wei, H.; Li, J.; Ni, H. DSNet: A Novel Way to Use Atrous Convolutions in Semantic Segmentation, 2024, [arXiv:cs.CV/2406.03702].

37. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep High-Resolution Representation Learning for Visual Recognition, 2020, [arXiv:cs.CV/1908.07919].
38. Qashqai, D.; Mousavian, E.; Shokouhi, S.B.; Mirzakuchaki, S. CSFNet: A Cosine Similarity Fusion Network for Real-Time RGB-X Semantic Segmentation of Driving Scenes, 2024, [arXiv:cs.CV/2407.01328].
39. Termritthikun, C.; Jamtsho, Y.; Ieamsaard, J.; Muneesawang, P.; Lee, I. EEEA-Net: An Early Exit Evolutionary Neural Architecture Search. *Engineering Applications of Artificial Intelligence* **2021**, *104*, 104397. https://doi.org/10.1016/j.engappai.2021.104397.