

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**SPORTA SACENSĪBU FOTOGRĀFIJU KLASIFICĒŠANA,
IZMANTOJOT DZIĻOS NEIRONU TĪKLUS
BAKALaura DARBS**

Autors: Vladislavs Gataļskis

Stud. apl. nr.: vg18027

Darba vadītājs: Dr.dat., profesors Kārlis Freivalds

RĪGA 2022

ANOTĀCIJA

Bakalaura darbā “Sporta sacensību fotogrāfiju klasificēšana, izmantojot dziļos neironu tīklus” tiek apskatīti mašīnmācīšanās veidi un visbiežāk izmantotā arhitektūra fotogrāfiju klasificēšanā – konvolucionālais neironu tīkls (CNN) un uz tā bāzēti algoritmi un ietvari. Kā arī darba ietvaros tika izstrādāts mežā notiekoša maratona fotogrāfiju kolekcijas augsta līmeņa klasifikators, kurš spēj noteikt fotogrāfijās attēlotos notikumus, balstoties uz atrasto cilvēku skaitu, seju skaitu un fotogrāfijā attēloto atrašanās vietu.

Mūsdienās gandrīz jebkuru fotogrāfiju klasificēšanas uzdevumu veic ar dziļās mācīšanās un dziļo neironu tīklu palīdzību, jo dziļā mācīšanās ļauj datoram iemācīties daudz sarežģītākas objektu iezīmes un veikt specifiskāku klasificēšanu.

Augsta līmeņa fotogrāfiju klasifikatori spēj noteikt informāciju, kas nav tikai fotogrāfijā redzami objekti. Tā informācija nav tieši attēlota, taču tiek izsecināta, balstoties uz vairākiem faktoriem.

Atslēgas vārdi: mašīnmācīšanās, dziļā mācīšanās, konvolucionālais neironu tīkls, klasificēšana, fotogrāfijas, attēlu segmentēšana.

ABSTRACT

CLASSIFICATION OF SPORT COMPETITION PHOTOGRAPHS USING DEEP NEURAL NETWORKS

The bachelor's thesis " Classification of sport competition photographs using deep neural networks" examines the types of machine learning and the most used architecture for classifying photographs - the convolutional neural network (CNN), as well as some popular algorithms and frameworks that are based on CNN. Also, a high-level classifier of the forest marathon photograph collection was developed, which can determine the events depicted in the photographs based on the number of people, the number of faces and the location in the photograph.

Today, almost any image classification task is performed using deep learning and deep neural networks because deep learning allows the computer to learn more complex object features and perform more specific classification.

High-level image classifiers can identify information that is not displayed in the image. This information is inferred from several factors.

Keywords: machine learning, deep learning, convolutional neural network, classification, photographs, image segmentation.

SATURS

APZĪMĒJUMU SARAKSTS	6
IEVADS.....	7
1. MAŠĪNMĀCĪŠANĀS PIELIETOŠANA FOTOGRAFĪJU KLASIFICĒŠANAS PROBLĒMAI	9
1.1 Mašīnmācīšanās veidi	9
1.2. Izmantotie mašīnmācīšanās veidi fotogrāfiju klasificēšanā	10
1.3. Dziļā mašīnmācīšanās un dziļie neironu tīkli.....	10
2. ATTĒLU ATPAZĪŠANAS UZDEVUMI UN SEGMENTĒŠANAS VEIDI.....	12
2.1 Attēlu atpazīšanas uzdevumi.....	12
2.1.1. Parastā attēlu klasifikācija.....	12
2.1.2. Attēlu tagošana	12
2.1.3. Attēlu klasificēšana ar lokalizēšanu.....	12
2.1.4. Objektu noteikšana	12
2.1.5. Attēlu segmentēšana	13
2.2. Attēlu segmentēšanas veidi	13
3. VISIZPLATĪTĀKĀ ATTĒLU KLASIFICĒŠANAS METODE “CNN” UN UZ TĀS BĀZĒTI POPULĀRI ALGORITMI UN IETVARI	15
3.1. Uz reģioniem bāzēts konvolucionālais neironu tīkls jeb R-CNN	16
3.1.1. Fast R-CNN	17
3.1.2. Faster R-CNN	17
3.1.3. Mask R-CNN	17
3.2. YOLO.....	18
4. AUGSTA LĪMEŅA KLASIFIKATORA IZSTRĀDE	19
4.1. Sporta sacensību klasificēšanas veidi un metodes	19
4.2. Izvēlētā datu kopa un tās definēšana	21
4.3. Izvēlētais ietvars klasifikatora izveidei (Mask R-CNN)	28
4.3.1. Izvēlētais attēlu atpazīšanas uzdevums	28
4.3.2. Izvēlētais segmentēšanas veids un attēlu segmentēšanas ietvars	28

4.3.3. Mask R-CNN implementācijas uzstādīšanas process	30
4.3.4. Iepriekš uztrenēts COCO modelis.....	31
4.4. Treniņdatu sagatavošanas process.....	32
4.4.1. Anotētie treniņdatu piemēri	34
4.5. Instanču segmentēšanas modeļa trenēšanas process	37
4.5.1. Trenēšana uz personīgā datora un Google Colab vidē.....	38
4.5.2. Treniņdatu sadalīšana apakškopās un hiperparametru noteikšana	39
4.5.3. Trenēšana uz visas anotētās datu kopas (96 attēli)	41
4.6. Uztrenētā instanču segmentēšanas modeļa testēšana un rezultāti	42
4.6.1. Iepriekšapmācīta modeļa instanču segmentēšanas salīdzināšana ar darba autora uztrenēto modeli.....	42
4.6.2. Rezultāti no testa kopas	44
4.6.3. Rezultāti no citas maratona fotogrāfiju kolekcijas.....	48
4.7. Fotogrāfijā attēlotā notikuma klasificēšanas algoritms	51
4.7.1. Klasificēšanas algoritma darbības apraksts.....	51
4.8. Klasifikatora testu rezultāti	55
4.8.1. Izvēlētās fotogrāfiju kolekcijas klasificēšanas testēšanas rezultāti.....	55
4.8.2. Citas, pēc satura līdzīgas fotogrāfiju kolekcijas testēšanas rezultāti	56
SECINĀJUMI.....	58
IZMANTOTĀ LITERATŪRA UN AVOTI	61
PIELIKUMI	65
1. pielikums: Augsta līmeņa klasificēšanas funkcijas kods.....	65

APZĪMĒJUMU SARAKSTS

ANN (angliski: *Artificial Neural Network*) – Mākslīgais neironu tīkls.

DNN (angliski: *Deep Neural Network*) – Dziļlais neironu tīkls

CNN (angliski: *Convolutional Neural Network*) – Konvolucionālais neironu tīkls.

R-CNN (angliski: *Region-Based Convolutional Neural Network*) – Uz reģioniem balstīts konvolucionālais neironu tīkls.

ROI (angliski: *Region of Interest*) – Interesējošai reģions jeb reģions, kurā, iespējams, ir vēlamais objekts.

RPN (angliski: *Region Proposal Network*) – Neironu tīkls reģionu priekšlikumu ģenerēšanai.

TL (angliski: *Transfer Learning*) – Mašīnmācīšanās tehnika, kurā tiek izmantots iepriekš apmācīts modelis, lai uztrenētu jaunu modeli cita, taču līdzīga uzdevuma veikšanai.

FT (angliski: *Fine Tuning*) – TL tehnika, kurā tiek atsaldēti iepriekšapmācīta modeļa slāņi, lai pielāgotu tā parametrus jaunai datu kopai.

SPE (angliski: *Steps Per Epoch*) – Mašīnmācīšanās trenēšanas procesa hiperparametrs, kas nosaka soļu jeb iterāciju skaitu vienā mācīšanās ciklā.

BS (angliski: *Batch Size*) – Mašīnmācīšanās trenēšanas procesa hiperparametrs, kas nosaka izmantoto trenēšanas paraugu skaitu vienā solī jeb iterācijā).

LR (angliski: *Learning Rate*) – Mašīnmācīšanās trenēšanas procesa hiperparametrs, kurš kontrolē to, cik lielā mērā tiek izmainīti neironu tīkla svāri, ņemot vērā novērtēto kļūdu.

CPU (angliski: *Central Processing Unit*) – Centrālais procesors.

GPU (angliski: *Graphics Processing Unit*) – Grafiskais procesors.

IEVADS

Fotogrāfiju klasificēšana balstās uz tiem pašiem principiem, kas piemīt objektu klasificēšanas procesam ikdienā. Cilvēki ikdienā veic klasificēšanu neskaitāmas reizes, jo objektu klasificēšana rada saprašanu par apkārt notiekošo, saprašanu par apkārt esošiem objektiem un to savstarpējo mijiedarbību.

Pēc būtības objektu, tai skaitā arī fotogrāfiju klasificēšana, nav pārāk sarežģīts uzdevums, ir nepieciešams tikai noteikt konkrētu objektu vai fotogrāfijā redzamo objektu iezīmes un, balstoties uz jau pieejamajām zināšanām par kategorijām, “jāievieto” objekts zem tās kategorijas, kura satur objektus ar klasificējamā objekta iezīmēm. Cilvēki šo procesu veic neaizdomājoties, jo tas ir kognitīvs process, bet kā šo uzdevumu varētu veikt dators, kuram nav iepriekšējo zināšanu par apkārtējo vidi un tajā esošajiem objektiem?

Datorprogrammas specifisku uzdevumu veikšanai ļoti bieži tiek rakstītas ar tradicionālās programmēšanas pieeju. Tas ir, programmētājs uzraksta algoritmu jau zināmiem ieejas datiem un veic ieejas datu apstrādi, lai iegūtu izvades datus kādā noteiktā formā. Lai izveidotu fotogrāfiju klasifikatoru ar tradicionālās programmēšanas pieeju, būtu jāvelta neaptverami ilgs laiks, lai sarakstītu algoritmā visus iespējamus gadījumus, objektu atrašanās vietas fotogrāfijā u.tml.; patiesībā tas pat nebūtu racionāls veids, kā šo uzdevumu veikt. Lai efektīvi izveidotu precīzu fotogrāfiju klasifikatoru, tiek izmantota mašīnmācīšanās pieeja.

Mašīnmācīšanās ir process, kurā dators pats mācās veikt kādu specifisku uzdevumu. Šī procesa rezultātā uztrenētais jeb apmācītais mašīnmācīšanās modelis spēj prognozēt vērtības vai klasificēt datus, balstoties pieredzē – apmācītā modeļa atgrieztā atbilde uz konkrētiem ieejas datiem nav tieši ieprogrammēta kodā.

Bakalaura darba ietvaros tika apskatīti mašīnmācīšanās veidi, attēlu atpazīšanas uzdevumi un segmentēšanas veidi, kā arī populārākā fotogrāfiju klasificēšanā izmantotā arhitektūra CNN un uz tās bāzēti populāri algoritmi un ietvari. Tika izvēlēta un definēta fotogrāfiju kolekcija, kura attēlo mežā notiekošu maratonu, lai noteiktu, kādi objekti ietekmē dotās fotogrāfiju kolekcijas klasificēšanu. Tika arī izvēlēts viens konkrēts uz CNN bāzēts ietvars “Mask R-CNN”, ar kura palīdzību tika izstrādāts izvēlētās fotogrāfiju kolekcijas augsta līmeņa klasifikators.

Bakalaura darba mērķis ir apskatīt CNN arhitektūru un uz tās bāzētus variantus, noteikt, kurš no veidiem vislabāk derētu izvēlētās fotogrāfiju kolekcijas klasificēšanai un, vai ir iespējams uztrenēto modeli (izveidoto klasifikatoru) pielietot citas, taču līdzīgas fotogrāfiju kolekcijas klasificēšanai.

Lai sasniegtu bakalaura darbā izvirzīto mērķi, pēc datu kopas definēšanas tika izveidota treniņkopa ar manuāli marķētiem datiem, noteikti optimālie trenēšanas procesa hiperparametri, izmantots uz COCO datukopas iepriekšapmācīts instanču segmentēšanas modelis, kurš tika pielāgots izvēlētajai fotogrāfiju kolekcijai ar TL pieejas palīdzību, un uzrakstīts klasificēšanas algoritms, kurš apstrādā instanču segmentēšanas modeļa atgrieztos rezultātus. Beigās tika veikta apmācītā instanču segmentēšanas modeļa un uzrakstītā klasificēšanas algoritma testēšana uz izvēlētajai fotogrāfiju kolekcijai un citai, līdzīgai fotogrāfiju kolekcijai.

1. MAŠĪNMĀCĪŠANĀS PIELIETOŠANA FOTOGRAFĪJU KLASIFICĒŠANAS PROBLĒMAI

1.1 Mašīnmācīšanās veidi

Eksistē trīs mašīnmācīšanās veidi: uzraudzītā mašīnmācīšanās, neuzraudzītā mašīnmācīšanās un stimulētā mašīnmācīšanās.

Stimulētā mašīnmācīšanās [1,2,3] ir tāds mašīnmācīšanās veids, kurā dators mācās veikt kādu konkrētu uzdevumu, veicot nejausus mēģinājumus un kļūdoties. Šī mašīnmācīšanās veida specifika ir tāda, ka dators par pareizi veiktu darbību iegūst “apbalvojumu”, tādējādi dators tiek informēts, ka tas ir veicis pareizu darbību, vai sodu, kas informē datoru, ka tas ir veicis nepareizu jeb kļūdainu darbību. Datora mērķis dotajā mašīnmācīšanās veidā ir palielināt atlīdzību. Stimulētā mašīnmācīšanās visbiežāk tiek izmantota tādos gadījumos, kad datoram ir jāveic kāds konkrēts uzdevums, atrodoties vidē, kur ir iespējama mijiedarbība ar apkārt esošiem objektiem. Šis veids tiek izmantots, piemēram, datorspēļu mākslīgā intelekta izstrādē, robotikā, pašbraucošu automašīnu realizēšanā.

Neuzraudzītā mašīnmācīšanās [4,5] ir mašīnmācīšanās veids, kurā dators mācās veikt kādu uzdevumu, piemēram, prognozēt vai klasificēt datus, mācoties uz nemarkētiem (angliski: unlabeled) datiem. Tas nozīmē, ka dators nezina, kas datus ir attēlots. Lai dators saprastu datu kontekstu, tiek izmantoti speciāli grupēšanas algoritmi, kuri grupē datus, balstoties uz slēptām datu iezīmēm (angliski: hidden patterns); modeļa trenēšanas laikā nav nepieciešama cilvēka iejaukšanās, dators pats cenšas noteikt līdzības un atšķirības datus, kā arī datu struktūru. Neuzraudzītās mašīnmācīšanās trūkumi ir diezgan ilgs trenēšanas laiks un lielāks risks iegūt neprecīzas prognozes.

Uzraudzītā mašīnmācīšanās [6] ir mašīnmācīšanās veids, kurā dators mācās veikt kādu uzdevumu, mācoties uz marķētiem (angliski: labeled) datiem. Marķēti dati [7] ir tādi dati, kuri ir tagoti (angliski: tagged) jeb apzīmēti ar vienu vai vairākiem tagiem jeb birkām, kuras identificē kādas konkrētas iezīmes, raksturlielumus, klasificēšanas kategorijas vai attēlotos objektus. Marķētie dati palīdz mašīnmācīšanās modelim saprast, vai veiktās prognozes un to varbūtības ir tuvu patiesībai vai nē. Neironu tīklu gadījumā modelis sākumā prognozē vērtību uz ieejas datiem, un pēc tam salīdzina prognozēto vērtību ar patieso ieejas datu vērtību. Lai noteiktu prognozēšanas kļūdu, tiek izmantota kāda no zaudējuma funkcijām, un kad ir zināma kļūda, neironu tīklā tiek izmainīti svāri, lai mēģinātu uzlabot prognozēšanas precizitāti jeb samazināt kļūdu. Dotais process notiek tik ilgi, kamēr kļūda nav maksimāli samazināta.

1.2. Izmantotie mašīnmācīšanās veidi fotogrāfiju klasificēšanā

Katram mašīnmācīšanās veidam ir sava specifika, kura jāņem vērā vēlamā uzdevuma risinājuma izstrādē.

Tā kā stimulētā mašīnmācīšanās pārsvarā tiek lietota gadījumos, kad nepieciešamais uzdevums jārisina, balstoties uz apkārtējo vidi un mijiedarbojoties ar tajā esošiem objektiem, fotogrāfiju klasificēšanā šis mašīnmācīšanās veids tiek pieminēts reti. Attēlu klasificēšanā tiek izmantoti tādi mašīnmācīšanās veidi kā uzraudzītā un neuzraudzītā mašīnmācīšanās.

Zinātniskajā rakstā “Comparative Analysis of Unsupervised and Supervised Image Classification Technique” [8] tika apskatītas un salīdzinātas uzraudzītās un neuzraudzītās mašīnmācīšanās metodes attēlu klasificēšanai, kā arī tika dokumentēti vērtīgi testu rezultāti.

Darbā tika apskatīti 2 uzraudzītās mašīnmācīšanās algoritmi (“Maximum Likelihood / Bayesian” un ANN) un 2 neuzraudzītās mašīnmācīšanās algoritmi (“K-means Clustering” un “Expectation-Maximization” (EM)). Visi 4 algoritmi tika implementēti, lai veiktu klasificēšanas testus. No visiem 4 algoritmiem tika secināts, ka viens uzraudzītās mašīnmācīšanās algoritms (ANN) un viens neuzraudzītās mašīnmācīšanās algoritms (“EM (Multivariate normal distribution)”) sasniedza labus rezultātus. Abi algoritmi veica klasificēšanu ar salīdzinoši augstu klasificēšanas precizitāti (1.testā: ANN – 67.12% un “EM (Multivariate normal distribution)” – 56.90%; 2. testā: ANN – 97.34% un “EM (Multivariate normal distribution)” – 89.26%), taču klasificēšanas procesa laiks bija daudz ātrāks ANN (1.testā: ANN – 796.98 sek. un “EM (Multivariate normal distribution)” – 2990 sek.; 2. testā: ANN – 801.57 sek. un “EM (Multivariate normal distribution)” – 3100 sek.).

No zinātniskā raksta “Comparative Analysis of Unsupervised and Supervised Image Classification Technique” var secināt, ka neironu tīkli ir ļoti noderīgi attēlu klasificēšanā, jo tie spēj veikt klasificēšanu ar diezgan augstu precizitāti un veic klasificēšanu salīdzinoši īsā laikā.

1.3. Dziļā mašīnmācīšanās un dziļie neironu tīkli

Dziļā mašīnmācīšanās [9,10] ir mašīnmācīšanās sfēras apakškopa, kurā tiek izmantoti neironu tīkli ar vismaz 2 neredzamajiem slāņiem (angliski: hidden layers). Tādi neironu tīkli simulē cilvēku smadzeņu darbību jeb uzvedību, kas ļauj datoram mācīties dažādas lietas līdzīgi kā to dara cilvēki. Lai gan neironu tīkli ar vienu neredzamo slāni joprojām spēj veikt aptuvenas prognozes (neironu tīkli ar 1 neredzamo slāni tiek saukti par mākslīgajiem neironu tīkliem jeb ANN), vairāki neredzamie slāņi (ANN ar vismaz 2 neredzamajiem slāņiem tiek uzskatīti par dziļajiem neironu tīkliem jeb DNN) var palīdzēt uzlabot un optimizēt prognožu precizitāti.

Dziļā mācīšanās tiek izmantota daudzos mākslīgā intelekta pielietojumos; tā uzlabo automatizāciju, analītisko un fizisko uzdevumu veikšanu bez cilvēka iejaukšanās.

Dziļā mācīšanās atšķiras no tradicionālās mašīnmācīšanās ar datiem, ar kuriem notiek darbošanās un ar metodē, kuras tiek izmantotas mācīšanās procesā. Tradicionālās mašīnmācīšanās algoritmi darbojas ar strukturētiem un marķētiem datiem, lai veiktu prognozes. Tas nozīmē, ka no ievaddatiem tiek definētas konkrētas iezīmes, lai šīs iezīmes varētu pārveidot modelim nepieciešamā strukturētā formā. Šis process tiek saukts pirmsapstrādi. Dziļā mācīšanās likvidē daļu no pirmsapstrādes procesa, kura parasti notiek tradicionālajā mašīnmācīšanās. Dziļās mācīšanās algoritmi spēj ielasīt un apstrādāt tādus nestrukturētus datus kā tekstu un attēlus, kā arī no nestrukturētiem datiem izgūt nepieciešamās iezīmes, tādējādi samazinot atkarību no cilvēka.

2. ATTĒLU ATPAZĪŠANAS UZDEVUMI UN SEGMENTĒŠANAS VEIDI

2.1 Attēlu atpazīšanas uzdevumi

Attēlu atpazīšana (angliski: image recognition) [11] ir datorredzes un mākslīgā intelekta apakš kategorija. Tā ir attēlu analizēšanas un objektu noteikšanas metožu kopums, kura iespējo dažādu specifisku uzdevuma automatizāciju.

Attēlu atpazīšana nodrošina iespēju identificēt atrašanās vietas, cilvēkus un dažādus citus attēlotos elementus, no kuriem ir iespējams veikt informatīvus secinājumus, analizējot attēlus.

Attēlu atpazīšana ietver sevī vairākus dažāda veida uzdevumus, kuri ir aprakstīti zemāk.

2.1.1. Parastā attēlu klasifikācija

Parastā attēlu klasifikācija ir visvienkāršākais attēlu klasificēšanas veids. Dotā klasificēšanas veida modelis sagaida attēlu, kurā ir tikai viens galvenais objekts. Modelis nosaka attēlā atrastās iezīmes un, balstoties uz zināšanām, prognozē, kurai klasei pieder attēlā redzamais objekts.

2.1.2. Attēlu tagošana

Attēlu tagošana [11] arī ir klasificēšanas uzdevums, taču dotais uzdevums veic precīzāku klasificēšanu nekā parastā attēlu klasificēšana. Attēlu tagošana spēj atpazīt vairāku konceptu vai objektu esamību attēlā, tādējādi piešķirot attēlam vienu vai vairākus tagus jeb birkas.

2.1.3. Attēlu klasificēšana ar lokalizēšanu

Attēlu klasificēšana ar lokalizēšanu [12], tā pat kā parastā attēlu klasificēšana, sagaida attēlu ar vienu objektu, kuru nepieciešams klasificēt. Taču atšķirība ir tāda, ka mašīnmācīšanās modelim jānosaka ne tikai klase, kurai pieder attēlā redzamais objekts, bet arī objekta atrašanās vieta attēlā. Objekta atrašanās vieta attēlā tiek iezīmēta ar taisnstūri jeb ierobežojošo kasti (angliski: bounding box), kas cieši aptver attēloto objektu. Tādējādi modelis, kurš veic šāda veida klasificēšanu, izdod nevis vienu, bet divas izvades: klasi un ierobežojošā loga (kastes) parametrus, piemēram, ierobežojošās kastes centra koordinātes x un y , augstumu un platumu.

2.1.4. Objektu noteikšana

Objektu noteikšanas [12] process veic vairāku objektu klasificēšanu un lokalizēšanu, un objekti var piederēt dažādām klasēm. Tas ir, attēlā tiek atrasti vairāki objekti un katram no tiem tiek piešķirta sava klase un iezīmēta sava ierobežojošā kaste.

2.1.5. Attēlu segmentēšana

Attēlu segmentēšana [13] ir uzdevums, kura mērķis ir sadalīt digitālo attēlu vairākos segmentos, lai samazinātu attēla sarežģītību un padarītu turpmāko attēla analizēšanu un apstrādi vienkāršāku.

2.2. Attēlu segmentēšanas veidi

Pastāv 3 attēlu segmentēšanas veidi: semantiskā segmentēšana, instanču segmentēšana un panoptiskā (angliski: “panoptic”) segmentēšana.

Semantiskā segmentēšana [14] ir attēlu segmentēšanas veids, kurā katrs attēla pikselis tiek klasificēts zem kādas no definētajām kategorijām. Tādējādi visas instances, kuras pieder konkrētai kategorijai, saņem vienu un to pašu klases iezīmi, kas nozīmē, ka visas instances ir atdalītas no attēla kā viena vesela daļa. Piemēram, ja attēlā ir redzams cilvēku pūlis, tad visi pikseļi, kuros ir redzami cilvēki, tiks saklasificēti zem kategorijas “cilvēks” un tādējādi semantiskās segmentēšanas modelis atgriezīs vienu klases “cilvēks” masku jeb iezīmēto laukumu. No šīs maskas nevar noteikt, cik cilvēku atrodas pūlī vai kuri cilvēki atrodas priekšplānā un kuri tālumā; ļoti maz detalizētas informācijas par attēlu.

Semantiskā segmentēšana ir ļoti noderīga nesaskaitāmu objektu klasificēšanai, piemēram, ja nepieciešams klasificēt kā atsevišķu attēla daļu debesis vai zāli. Debesis vai zāle ir nesaskaitāmas lietas, tās vairāk apraksta, piemēram, attēla fonu, tāpēc šajā gadījumā netiek zaudēta kāda detalizēta informācija vai konteksts.

Instanču segmentēšana [14] ir tāds attēlu segmentēšanas veids, kas apstrādā vairākus objektus, kuri pieder vienai un tai pašai klasei, kā atšķirīgus objektus, nevis kā vienu veselu kopumu, kā tas notiek semantiskajā segmentēšanā. Katrai klases instancei tiek iezīmēta sava ierobežojošā kaste un objekta segmentēšanas maska, kura nosaka konkrētās instances robežas jeb kontūru. Piemēram, ja attēlā ir redzams cilvēku pūlis, tad instanču segmentēšanas modelis atrastu, iezīmētu segmentēšanas masku un klasificētu katru cilvēka instanci pūlī. Tādējādi var iegūt gan konkrētai klasei piederošo atrasto objektu skaitu, gan konkrētas instances atrašanās vietu attēlā.

Instanču segmentēšanas specifika dod vairāk detalizētas informācijas par fotogrāfijā attēloto.

Panoptiskā segmentēšana [14,15] parādījās salīdzinoši nesen. Pirmo reizi ar šo segmentēšanas veidu iepazīstināja Kirillov et al. 2018. gadā. Panoptiskā segmentēšana apvieno abus augstāk minētos segmentēšanas veidus: semantisko un instanču segmentēšanu.

Panoptiskajā segmentēšanā instances var būt divu veidu: instances, kuras attēlo saskaitāmus objektus jeb lietas (angliski: things), un instances, kuras attēlo nesaskaitāmus objektus (angliski: stuff).

Panoptiskā segmentēšana nodrošina vairāk konteksta nekā instanču segmentēšana un ir daudz detalizētāka nekā semantiskā segmentēšana. Šīs īpašības padara panoptisko segmentēšanu ļoti noderīgu tādos mašīnmācīšanās uzdevumos, kur ir nepieciešama apkārtējās vides izpratne.

3. VISIZPLATĪTĀKĀ ATTĒLU KLASIFICĒŠANAS METODE “CNN” UN UZ TĀS BĀZĒTI POPULĀRI ALGORITMI UN IETVARI

Mašīnmācīšanā tiek izmantoti dažādi klasificēšanas algoritmi [16], piemēram, “SVM” (Support Vector Machine), “K-nearest neighbor”, “random forest” u.tml., taču šie algoritmi neprot darboties ar attēliem, bet gan tikai ar skaitliskiem datiem. Tāpēc, lai izmantotu šos algoritmus attēlu klasificēšanai, sākumā nepieciešams no attēliem “izgūt” svarīgākās iezīmes, kuras apraksta objektus, un pēc tam jau šīs iezīmes var padot tradicionālajiem klasificēšanas algoritmiem, kā “SVM” vai “random forest”.

Konvolucionālais neironu tīkls jeb CNN [17,18] ir viens no vairākiem mākslīgo neironu tīklu (ANN) veidiem, kas visbiežāk tiek izmantots attēlu analizēšanas un apstrādes uzdevumos. CNN sastāv no ievades slāņa, izvades slāņa un neredzamajiem slāņiem (angliski: hidden layers), turklāt neredzami slāņi ietver sevī arī konvolucionālos slāņus, apvienošanas slāņus (angliski: pooling layers), pilnībā savienotus slāņus (angliski: fully connected layers) un normalizācijas slāņus (angliski: normalization layers). Konvolucionālie slāņi ir tie slāņi, kuri ir atbildīgi par specifisku iezīmju “izgūšanu” no attēliem. Konvolucionālie slāņi satur filtrus jeb iezīmju detektorus, kuri ir reprezentēti kā mazas $N \times N$ izmēru matricas. CNN padotajam attēlam tiek pielietota konvolūcijas operācija ar M šādām matricām jeb filtriem, lai iegūtu M izvades jeb M iezīmju kartējumus (angliski: feature maps). Kad iezīmju kartējums tiek izveidots, katra vērtība kartējumā tiek izlaista cauri ReLU aktivācijas funkcijai. Pēc iezīmju kartējumu iegūšanas tiek veikta apvienošanas operācija (visbiežāk tā ir “max pooling” operācija), lai samazinātu iezīmju kartējuma matricas izmēru, tādējādi samazinot apgūstamos parametrus (angliski: learnable parameters) un neironu tīklā veikto aprēķinu apjomu, kā arī apvienošanas slāņi palīdz noteikt konkrētās iezīmes esamību nevis tiešo iezīmes atrašanās vietu padotajā attēlā. Tas padara modeli izturīgāku pret ieejas attēla iezīmju novietojuma izmaiņām [19]. Pēc apvienošanas operācijas jeb pēc apvienošanas slāņa var atkal sekot konvolucionālais slānis, izmantota ReLU aktivācijas funkcija un apvienošanas slānis, bet var arī sekot saplacināšanas (angliski: flatten) slānis jeb operācija. Saplacināšanas operācija visus iegūtos (pēc apvienošanas operācijas) iezīmju kartējumus saliek vienā, garā vektorā, lai to varētu padot CNN klasificēšanas daļai jeb pilnībā savienotam neironu tīklam.

CNN ir ļoti populāra attēlu klasificēšanas metode, jo tā spēj iemācīties ne tikai klasificēt fotogrāfijas, bet arī iemācīties iezīmju izgūšanu mācīšanās laikā. Pirms CNN un dziļās mācīšanās, lai veiktu attēlu klasificēšanu, cilvēki veica iezīmju izgūšanu manuāli un tad iezīmes tika padotas tradicionālajiem klasificēšanas algoritmiem kā ieejas dati [20]. CNN, savukārt, spēj šos divus procesus apvienot vienā.

CNN ir ļoti labs rīks attēlu klasificēšanai un objektu noteikšanai ar ierobežojošām kastēm, taču CNN nespēj darboties ar vairākiem objektiem vienlaikus; tas darbojas labi tikai tad, ja attēlā ir tikai viens galvenais objekts. Lai šo problēmu atrisinātu, tika izstrādāti ietvari un algoritmi, kuri izmanto CNN kā vienu no sastāvdaļām.

3.1. Uz reģioniem bāzēts konvolucionālais neironu tīkls jeb R-CNN

Uz reģioniem bāzēts konvolucionālais neironu tīkls jeb R-CNN (angliski: Region based CNN) [21,22] ir papildinātais CNN variants. Tas tiek izmantots gan attēlu klasificēšanai, gan objektu noteikšanai. R-CNN balstās uz tāda koncepta kā reģionu priekšlikumi (angliski: region proposals); priekšnoteikums ir tāds, ka katrā reģionā dominē tikai viens vienīgs interesējošs objekts (angliski: object of interest) jeb ROI. R-CNN izmanto tādu meklēšanas algoritmu kā “selective search”, lai uzģenerētu reģionu priekšlikumus. Reģionu priekšlikumi ir reģionu kopa, kur katrs reģions, iespējams, satur objektu.

R-CNN darbojas pēc sekojoša principa: sākumā ar “selective search” algoritma palīdzību tiek uzģenerēti aptuveni 2000 reģionu priekšlikumi un katrs reģions tiek “iesaiņots” (angliski: wrapped), lai to būtu iespējams padot neironu tīklam (tā kā CNN ieejā sagaida fiksēta izmēra attēlus, katrs reģions ir “jāiesaiņo” jeb jāizmaina reģiona izmērs uz tādu attēla lielumu, ko sagaida ieejā CNN). Tad katrs “iesaiņotais” reģions tiek padots CNN; CNN šajā gadījumā darbojas kā iezīmju “izgūvējs” (angliski: feature extractor), kas iegūst fiksēta izmēra vektoru no katra reģiona. Kad iezīmes no reģiona ir “izgūtas” ar CNN palīdzību, tiek pielietots SVM algoritms, kas veic reģionā esoša objekta klasificēšanu, un ierobežojošās kastes regresors (angliski: bounding-box regressor), kas prognozē 4 vērtības, kuras apraksta ierobežojošās kastes atrašanās vietu un izmēru. Var gadīties, ka objektu noteikšanas modelis atradīs vienu un to pašu objektu vairākas reizes (vienam objektam vairākas “noteikšanas” (angliski: multiple detections)), taču šādi gadījumi nav pieņemami, tāpēc, kad visiem reģioniem tika noteiktas ierobežojošās kastes, tiek pielietota NMS jeb “Non-Maximum suppression” tehnika [23]. NMS nodrošina konkrēta objekta noteikšanu tikai vienreiz.

R-CNN lielākais trūkums ir modeļa trenēšanas laiks. Tas ir patiešām lēns, jo tiek ģenerēti 2000 reģioni ar “selective search” algoritma palīdzību, katrs reģions tiek laists cauri CNN, lai izgūtu iezīmes (tātad 2000 reizes tiek pielietots CNN), un, kopumā skatoties, R-CNN objektu noteikšanas metode izmanto 3 dažādus modeļus: CNN – iezīmju izgūšanai, SVM – objektu jeb reģionu klasificēšanai un ierobežojošās kastes regresors – ierobežojošās kastes parametru noteikšanai. Šie trūkumi tika ātri noteikti un tika veikti mēģināji uzlabot R-CNN arhitektūru.

3.1.1. Fast R-CNN

Fast R-CNN [24] ir uzlabotais R-CNN modelis. Atšķirībā no R-CNN, kas laiž cauri CNN katru reģionu (~2000 reizes), Fast R-CNN laiž cauri CNN visu attēlu tikai vienu reizi, lai izgūtu iezīmes. Joprojām tiek izmantots “selective search” algoritms, lai uzģenerētu ~2000 reģionu priekšlikumus, taču katrs reģions tiek nevis padots CNN, bet tiek projicēts uz iegūtajiem iezīmju kartējumiem. Objektu noteikšanā gandrīz katram reģionu priekšlikumam būs atšķirīga forma jeb izmērs, bet tā kā pilnībā savienotie slāņi (angliski: fully connected layers) sagaida fiksēta izmēra vektorus jeb ievadi, jo neironu tīklam ir piešķirts fiksēts svaru skaits, tika ieviests tāds process kā “ROI Pooling” jeb interesējošo reģionu apvienošana [25], kas ļauj visus interesējošos reģionus jeb ROI pārveidot uz fiksētu izmēru jeb formu. Interesējošo reģionu apvienošana tiek pielietota pēc pēdējā konvolucionālā slāņa, izmetot pēdējo parasto, piemēram, “max-pooling” apvienošanas slāni.

Kopsavilkumā, Fast R-CNN neizmanto SVM klasifikatoru, bet gan “Softmax”. Ierobežojošās kastes prognozēšanai tiek izmantota zaudējuma funkcija “smooth L1 loss”. Tādējādi Fast R-CNN tiek trenēts vienā posmā, nevis trijos kā R-CNN, kur tika atsevišķi trenēti CNN, SVM un regresora modelis. Lielākais Fast R-CNN trūkums ir tas, ka reģionu priekšlikumu ģenerēšanai tiek izmantots “selective search” algoritms; šis algoritms padara reģionu priekšlikumu ģenerēšanas procesu lēnu un laikietilpīgu.

3.1.2. Faster R-CNN

Faster R-CNN [26] ir uzlabotais Fast R-CNN modelis. Dotā modeļa specifika ir tāda, ka, lai ģenerētu reģionu priekšlikumus, tiek izmantots nevis “selective search” algoritms, bet gan atsevišķs neironu tīkls – RPN. Iezīmju kartējumi no CNN tiek padoti RPN, lai uzģenerētu reģionu priekšlikumus.

3.1.3. Mask R-CNN

Mask R-CNN [27], savukārt, ir papildinātais Faster R-CNN variants, kura uzdevums fokusējās nevis uz objektu noteikšanu attēlā, bet gan uz attēlu segmentēšanu, konkrētāk, instanču segmentēšanu. Mask R-CNN veic visu to pašu, ko Faster R-CNN, bet papildus atgriež katra atrastā objekta jeb instances segmentēšanas masku.

Mask R-CNN tika izstrādāts uz Faster R-CNN pamata. Kamēr Faster R-CNN atgriež 2 izvades katram kandidātobjektam – objekta kategoriju un ierobežojošās kastes nobīdi jeb atrašanās vietas koordinātes –, Mask R-CNN izdod papildus vēl vienu izvadi – objekta segmentēšanas masku. Šī papildus funkcionalitāte jeb objekta maskas noteikšana notiek paralēli kategorijas un ierobežojošās kastes noteikšanai.

3.2. YOLO

YOLO jeb “You Only Look Once” [28,29] ir objektu noteikšanas algoritms, kas spēj nodrošināt reāla laika objektu noteikšanu. YOLO kļuva diezgan populārs precizitātes un galveno kārt ātrdarbības dēļ.

YOLO izmanto CNN, lai veiktu reāla laika objektu noteikšanu. Kā izriet no algoritma nosaukuma “You Only Look Once”, ir nepieciešams tikai vienu reizi iziet cauri neironu tīklam, lai noteiktu objektus. CNN tiek izmantots, lai noteiktu dažādu klašu varbūtības un ierobežojošās kastes vienlaicīgi.

YOLO algoritmam ir vairāki varianti. Visizplatītākie ir “Tiny-YOLO” un “YOLOv3”.

Pastāv divu tipu objektu detektori jeb objektu noteicēji: viena posma (angliski: one-stage) un divu posmu (angliski: two-stage) detektori. Divu posmu detektori ir tādi ietvari kā Fast R-CNN un Faster R-CNN, jo sākumā tie veic reģionu priekšlikumu ģenerēšanu (vai nu ar “selective search” algoritmu, vai nu ar atsevišķu neironu tīklu (RPN)), un tad klasificē šos reģionus. Savukārt, viena posma detektori, tādi kā YOLO vai SSD (“Single Shot Detector”), ir “proposal free” algoritmi jeb tiem nav nepieciešams ģenerēt reģionu priekšlikumus, lai atrastu objektus.

YOLO ir viena posma detektors. Tā vietā, lai ģenerētu reģionu priekšlikumus, YOLO sadala attēlu režģī ar N šūnām, kur katras šūnas izmērs ir $S \times S$. Katra šūna ir atbildīga par to objektu noteikšanu un lokalizēšanu, ko konkrētā šūna satur. Katra šūna prognozē ierobežojošās kastes koordinātes relatīvi pret savām koordinātēm, kā arī prognozē konkrētajā šūnā esošo objektu klases un varbūtības. Tā kā vairākas šūnas var atrast vienu un to pašu objektu, rodas daudz dublikātu. Lai šo problēmu atrisinātu, YOLO izmanto NMS tehniku, lai atstātu tikai to objekta noteikšanu (angliski: single object detection), kurai ir vislielākā varbūtība.

YOLO algoritmam ir arī savi trūkumi. YOLO ir grūtības ar mazu objektu noteikšanu, ja tie atrodas grupās, jo katra šūna var noteikt tikai vienu objektu. Kā arī YOLO precizitāte ir diezgan zema, salīdzinot ar daudz lēnāku objektu noteikšanas ietvaru kā Fast R-CNN.

4. AUGSTA LĪMEŅA KLASIFIKATORA IZSTRĀDE

Fotogrāfiju klasificēšana var būt vienkārša vai augsta līmeņa. Vienkārša līmeņa klasificēšana visbiežāk apzīmē tāda tipa klasificēšanu, kur attēlam tiek piešķirta viena vai vairākas klases, kuras apzīmē, piemēram, attēlā redzamos objektus. Augsta līmeņa klasificēšana ir tāda tipa klasificēšana, kurā attēls tiek klasificēts, balstoties uz vairākiem faktoriem, tai skaitā attiecības starp objektiem un kontekstu.

Dažādi pasākumi notiek katru dienu gandrīz visur pasaulē, daudzi no tiem tiek iemūžināti, lai būtu iespējams šo pasākumu parādīt tiem, kuri nebija uz vietas, vai vienkārši neļaut pasākumam tikt aizmirstam. Taču pasākumu fotogrāfiju kolekciju klasificēšana, tai skaitā sporta fotogrāfiju kolekciju klasificēšana, balstās uz pasākuma kontekstu jeb saturu, tas ir, dažādu attēloto objektu kombinācijas attēlā var paskaidrot vai identificēt kādu specifisku pasākuma notikumu.

Lai izveidotu augsta līmeņa fotogrāfiju klasifikatoru, nepieciešams noteikt attēlotos objektus jeb to klases, objektu skaitu un atrašanās vietu. Veicot iegūto rezultātu apstrādi, var identificēt specifiskus pasākuma notikumus, lai samazinātu, piemēram, specifisku notikumu attēlojošu fotogrāfiju meklēšanas laiku kolekcijā. Šādu klasifikatoru varētu izmantot, piemēram, dizaineri vai fotogrāfi. Mājaslapu dizainerim varētu būt aptuvenā dizaina ideja, kur viņš, piemēram, iedomājas divus cilvēkus blakus vairākām mašīnām; augsta līmeņa klasifikators varētu saklasificēt fotogrāfiju kolekciju, ar kuru dizainers strādā, un, izmantot filtrus, atlasīt tikai tās fotogrāfijas, kuras apmierina dizainera specifiskā notikuma prasības.

4.1. Sporta sacensību klasificēšanas veidi un metodes

Sporta sacensības ir process, kurā sportisti sacenšas, veicot kādas fiziskas aktivitātes vai spēles. Eksistē daudz dažādu sporta veidu, kur katrā ir savi noteikumi, veicamās aktivitātes, laukumi un atrašanās vietas, kurās notiek sacensības, un arī izmantotais inventārs.

Sporta sacensību fotogrāfijas var klasificēt vairākos veidos.

Var klasificēt fotogrāfijas, nosakot tikai sporta veidus, piemēram, noteikt, vai fotogrāfijā attēlotais sporta veids ir futbols, basketbols vai beisbols. Šis būtu visvienkāršākais klasificēšanas veids, jo tiek iegūta tikai vispārīga informācija par attēlu.

Var klasificēt fotogrāfijas, balstoties uz vispārīgām lietām, piemēram, uz attēloto cilvēku skaitu un to atrašanās vietu. Atrašanās vietas var arī būt specifiskas konkrētam sporta veidam.

Var klasificēt fotogrāfijas, balstoties uz konkrētā sporta veida aktivitātēm. Piemēram, noteikt, ka basketbolists met bumbu grozā vai vārtusargs atsargā sitienu pa vārtiem. Attēloto

cilvēku pozas var palīdzēt noteikt kādu konkrētu specifiska sporta veida darbību jeb aktivitāti, ko attēlotais cilvēks vai sportists veic.

Zinātniskajā rakstā “Real-time field sports scene classification using colour and frequency space decompositions” [30] tika veikta laukumu izmantojošo sporta sacensību/veidu (piemēram, futbols, amerikāņu futbols, beisbols, u.tml.) ainu klasificēšana, kuras kategorijas balstās ne tikai uz fotogrāfijā vai reāla-laika video kadrā attēlotajiem objektiem, bet arī uz ainas fotografēšanas rakursiem. Eksperimentu veikšanai tika definētas 14 dažādas klases, kuras reprezentē gandrīz visas iespējamās situācijas uz laukuma mača laikā. Viens no definētajiem klases piemēriem ir “tuvplāna kadrs galva vienkāršs fons” (angliski: “close up shot head simple background”), kas nosaka to, ka ir redzama cilvēka vai vairāku cilvēku galvas vai sejas tuvplānā; vienkāršs fons nozīmē to, ka fons attēlo viegli atšķiramu dominējošu krāsu (savukārt, sarežģīts fons (angliski: complex background) nozīmē, ka fonā nav redzama kāda viena dominējoša krāsa). Tātad fotogrāfijas var klasificēt arī pēc redzamās ainas tipa nevis tikai pēc attēlotā satura.

Sporta sacensību vai jebkādu cita pasākuma fotogrāfijā attēloto notikumu klasificēšana ir nedaudz līdzīga attēla aprakstīšanas uzdevumam (angliski: image captioning) [31] – konkrētajam attēlam tiek uzģenerēts paskaidrojošs teksts, kas apraksta attēlā redzamo notikumu. Visbiežāk šī uzdevuma risināšanai tiek izmantota kodēšanas un dekodēšanas (angliski: encoder-decoder) arhitektūra. Sākumā tiek izmantots CNN kā datu iekodētājs, lai izgūtu no attēla iezīmes, kuras reprezentē fotogrāfijā attēloto informāciju, un tad iegūtās iezīmes tiek padotas LSTM neironu tīklam, kurš spēj uzģenerēt fotogrāfijā attēloto teksta formā. Ar CNN-LSTM arhitektūras pieeju ir grūti veidot fotogrāfiju klasifikatoru, jo tā atgriež aprakstu par fotogrāfijā attēloto, kas var arī nebūt konkrēta kategorija, bet vienkārši attēlu aprakstošs teikums. Kā arī CNN-LSTM modelis nespēj noteikt konkrētu objektu skaitu vai to atrašanās vietu fotogrāfijā, kas samazina pieejamo detalizēto informāciju par fotogrāfijā attēloto.

Zinātniskajā rakstā “Event Recognition Based on Classification of Generated Image Captions” [32] tika piedāvāts risinājums, kurš, balstoties uz uzģenerētiem fotogrāfiju aprakstiem, veic fotogrāfijā attēlotā notikuma klasificēšanu. Piedāvātais risinājums arī izmanto iekodēšanas un dekodēšanas arhitektūru, taču ir arī papildus moduļi, kuri veic uzģenerēto aprakstu apstrādi, lai prognozētu fotogrāfijā attēloto notikuma kategoriju.

Tā kā ar attēlu segmentēšanas palīdzību no fotogrāfijas ir iespējams iegūt patiešām lielu detalizētās informācijas apjomu (atsevišķas objektu instances, segmentēšanas maskas, objektu skaits), tika pieņemts lēmums izmantot attēlu segmentēšanu modeli kā fotogrāfiju klasifikatora sastāvdaļu.

4.2. Izvēlētā datu kopa un tās definēšana

Izvēlētā datu kopa jeb fotogrāfiju kolekcija attēlo Latvijas čempionātu pagarinātajā distancē jeb maratonā [33]. Kolekcija satur 1172 fotogrāfijas un tās autors ir Ojārs Millers.

Maratons ir skriešanu sacensības, kurā dalībnieki skrien garu distanci. Maratoni parasti norisinās uz kādiem izvēlētiem ceļiem pilsētās, bet var arī norisināties mežā, kur dalībnieki skrien pa meža takām.

Lai noskaidrotu, pēc kādiem kritērijiem būtu iespējams klasificēt doto un tai līdzīgas fotogrāfiju kolekcijas, tika apskatītas visas izvēlētajā kolekcijā esošās fotogrāfijas, lai noteiktu, kuri objekti varētu palīdzēt identificēt konkrēto sacensību notikumu jeb, ko fotogrāfija attēlo.

Zemāk ir aprakstītas dažas pēc kategoriju veida atšķirīgas fotogrāfijas, kuras palīdz definēt klasificēšanas kategoriju veidus un kategorijas, kuras konkrētāk paskaidro iespējamo attēloto notikumu.



4.1. att. Cilvēku grupa no priekšas pasākuma stāvvietas zonā

Attēlā 4.1. ir redzami tādi saskaitāmie objekti kā cilvēki un automašīnas. Nesaskaitāmie objekti jeb fonu aprakstošie elementi ir mežs uz zālājs. Dotajā attēlā ir redzami vismaz 10 cilvēki, taču dominējošie šajā attēlā ir 4 cilvēki priekšplānā; fonā esošie cilvēki ir ļoti mazi un tie nav fokusā. Ir redzamas arī vismaz 9 mašīnas, kuras nav fokusā, bet tās var diezgan labi saredzēt. Šo 4 cilvēku grupu var uzskatīt par grupu, kas ir vērsta pret kameru, jo tikai 1 cilvēks ir vērsts ar muguru pret fotokameru jeb nav redzama cilvēka seja; ja ir redzama seja no sāna, tad cilvēks tomēr ir vairāk vērsts fotokameras virzienā nekā pret to. Liels automašīnu skaits attēlā var nozīmēt to, ka automašīnas atrodas ceļu satiksmē vai autosacīkstēs kustībā, bet var arī nozīmēt to, ka automašīnas nav kustībā jeb tās ir noparkotas stāvvietā. Tā kā maratons ir sacensības, kurās piedalās un galveno aktivitāti – skriešanu – veic cilvēki, tad liels automašīnu skaits dotajā fotogrāfiju kolekcijā apzīmētu automašīnu stāvvietas zonu.



4.2. att. Maza cilvēku grupa galvenajā pasākuma notikuma vietā

Pasākumos vienmēr ir tāda vieta, kurā norisinās vai nu pasākuma rīkošanas iemesla darbības, piemēram, dziedāšana koncertā, vai arī vieta, kurā notiek visvairāk dažādu aktivitāšu neatkarīgi no pasākuma veida. Dotajā fotogrāfiju kolekcijā galvenais notikums ir skriešanas sacensības, taču tās notiek gandrīz visās iespējamajās atrašanās vietās: mežā, zālājā, blakus finišam u.tml., savukārt, galvenā pasākuma notikuma vieta ir tā, kur atrodas visvairāk cilvēku un kur notiek dažādas “nesportiskas” aktivitātes. Tā ir vieta, kur atrodas nojumes jeb tentis, kurās, iespējams, tiek pārdoti suvenīri vai notiek pierēģistrēšanās. Tā ir vieta, kur cilvēki, t.sk. skatītāji, sarunājās, vēro apkārt notiekošo un labi pavada laiku. Tādi objekti kā pludmales banneri un nožogojuma banneri arī apraksta galveno pasākuma notikuma vietu, jo jebkāda tipa bannerus parasti novieto tur, kur atrodas daudz cilvēku, lai bannerus varētu saskatīt visi.

Attēlā 4.2. ir redzami 3 cilvēki. Kaut arī jebkurš cilvēks, skatoties uz šo fotogrāfiju, var nojaust, ka 2 cilvēki priekšplānā veido grupu vai pāri un 1 cilvēks fonā atrodas viens pats, objektu noteikšanas modelis redz 3 cilvēkus, kas varētu liecināt par to, ka, iespējams, ir redzama 3 cilvēku grupa. Objektu noteikšanas modelis spēj noteikt objektus un to atrašanos vietu attēlā jeb 2-dimensionālā telpā nevis 3-dimensionālā. Attēlā noteikt attēlojamās 3-dimensionālās telpas dziļumu ir pavisam cits un sarežģīts uzdevums.



4.3. att. Sacensību dalībnieki uz starta līnijas

Attēlā 4.3. ir attēloti sacensību dalībnieki, kuri gatavi startēt. Šajā attēlā ir redzami ļoti liels dalībnieku skaits, tāpēc dotajam attēlam viena no kategorijām būtu liela cilvēku grupa vai pat cilvēku pūlis. Ja klasificēt šo attēlu zem kategorijas “grupa”, tad klasifikators būs ļoti vispārīgs jeb kategorija “grupa” nozīmēs tikai to, ka attēlā tika atrasti vairāk par 2 cilvēkiem. Attēlā 4.3. var arī pamanīt, ka ir redzamas gan automašīnas, kuras identificē atrašanās vietu “stāvvietas zona”, gan 1 tenti, kura identificē atrašanās vietu “galvenā pasākuma notikuma vieta”. Gan tentis, gan automašīnas var būt fonā, tāpēc dotajā gadījumā būtu jāievieš prioritāšu sistēma jeb jānosaka, kuri objekti (ņemot vērā arī objektu skaitu) definē atrašanās vietu ar augstāku prioritātei jeb paskaidro daudz specifiskāku notikumu. Piemēram, ja ir atrasts mežs un vairākas automašīnas, tad prioritāte būs atrašanās vietas kategorijai “stāvvietas zona”, jo atrastās automašīnas dod vairāk konteksta nekā, ja būtu atrasts tikai mežs.



4.4. att. Čempionāta dalībnieks mežā

Attēlā 4.4. ir attēlots čempionāta dalībnieks mežā, kas ir vēl viena atrašanās vietas kategorija. Koki ir tie objekti, kuri vislabāk raksturo mežu, taču jāņem vērā, ka mežos uz zemes gandrīz vienmēr ir sūnas nevis zāle. Sūnu un zāles tekstūra atšķiras, bet, vai mašīnmācīšanās modelis spēs atšķirt sūnas no zāles? Iespējams, ka ar lielu treniņkopu, mašīnmācīšanās modelis spētu noteikt konkrētas sūnu un zāles iezīmes, tādējādi sūnas mežā netiktu iezīmētas un klasificētas kā zāle, taču darba izstrādē tomēr netika izmantota kategorija “sūnas” kaut arī tā var tikt izmantota, lai noteiktu atrašanās vietu “mežs”.



4.5. att. Čempionāta dalībnieki pļavā tuvojās mežam

Attēlā 4.5. var pamanīt, ka lielāko fotogrāfijas fona daļu aizņem plašs zāles laukums jeb zālājs, taču ir redzams arī meža sākums. Klasifikatoram būtu jābūt noteikt, kurā gadījumā cilvēki atrodas iekšā mežā un kurā gadījumā tie atrodas citā vietā.



4.6. att. Čempionāta dalībnieki apbalvošanas zonā

Attēlā 4.6. ir attēlota triju dalībnieku apbalvošana. Apbalvošanas zonu, kas ir vēl viena atrašanās vietas kategorija, raksturo pjedestāls, medaļas un bannera siena (angliski: banner wall). Bannera sienas bieži tiek novietotas apbalvošanas zonā, lai būtu redzami sacensību sponsori un partneri, taču tā kā dažādiem pasākumiem var būt dažādi sponsori (tas ir atkarīgs no sacensību veidu, atrašanās vietas, valsts, u.c.), tad dotais objekts identificē tikai šī vai šo organizatoru veidotas sacensības; iespējams, mašīnmācīšanās modelis nespēs noteikt bannera sienu, kura ir citā krāsā vai uz kuras ir attēloti citi sponsori. Medaļas arī identificē apbalvošanas procesu, taču medaļas ir salīdzinoši mazi objekti izmēros, kas varētu radīt grūtības ar lielu varbūtību noteikt, ka tika atrasta medaļa, kā arī uz medaļām tiek gravētas ļoti atšķirīgas lietas un teksti, kuras arī ir atkarīgas no sacensību veida un valsts; medaļas var būt arī ne tikai apaļas. Pjedestāli, savukārt, gandrīz visi ir ar līdzīgu formu, pakāpienu izkārtojumu un ciparu apzīmējumiem.



4.7. att. Organizators starp čempionāta dalībnieki

Attēlā 4.7. ir redzami sacensību dalībnieki, katram virsū ir lapiņa ar numuru. Apskatot attēlu, var secināt, ka dalībnieku formas krekli ir līdzīgi vai pat vienādi, kas identificētu piederību kādām izglītības iestādēm vai sporta organizācijām, taču maratonos dalībnieku formām lielas nozīmes nav. Katrs dalībnieks ir pats par sevi, viņu identificē tikai numurs uz lapiņas. Nosakot lapiņas esamību uz cilvēka varētu apzīmēt to, ka dotais cilvēks ir sacensību dalībnieks, taču tās ir salīdzinoši mazas, dažādos pasākumos šīm lapiņām var būt citādāks dizains un vairākos gadījumos šīs lapiņas pat nav redzamas, piemēram, kad cilvēks ir fotografēts no muguras (ja lapiņas piestiprina no priekšas) vai no sāna, tā var tikt aizsegta ar rokām utt. Pasākuma organizatoru forma ir vienīgā forma dotajā pasākumā, kura identificē konkrētu cilvēku grupu, jo tām ir vienādas krāsas un viens un tas pats uzraksts. Iemācīt identificēt organizatorus dotajā fotogrāfiju kolekcijā ir iespējams, bet tas nebūs pielietojams citām, līdzīgām fotogrāfiju kolekcijām, vienīgi, ja organizatori nebūs tie paši un formas netiks mainītas.

Izpētot un aprakstot izvēlēto fotogrāfiju kolekciju var secināt, ka tā attēlo vairākas pasākuma notikumu jeb tādas atrašanās vietas kā mežs, zālājs, galvenā pasākuma notikuma vieta, stāvvietas zona un apbalvošanas zona. Gandrīz visās fotogrāfijās ir saskatāma kategoriju veidu šķelšanās jeb apvienojums, piemēram, maza cilvēku grupa mežā – viens kategoriju veids nosaka cilvēku skaitu fotogrāfijā, bet otrs – cilvēku grupas atrašanās vietu; kombinējot kategoriju veidus, tiek iegūts precīzāks konkrēta notikuma apraksts. Cilvēku novietojums pret kameru arī ir viens kategoriju veidiem, balstoties uz kuru var klasificēt fotogrāfijas, lai iegūtu vēl detalizētāku informāciju par fotogrāfijā attēloto. Pēc izvēlētās fotogrāfiju kolekcijas izpētīšanas tika arī noskaidrots, ka maratonā gandrīz vai nav tādu objektu, kuri nosaka ar

dotajām sporta sacensībām saistītās aktivitātes, kā tas ir, piemēram, futbolā vai jebkādā citā sporta veidā, kur ir pieejams konkrēta sporta veida inventārs.

Izvēlētās maratona fotogrāfiju kolekcijas definēšanas procesā noteiktie kategoriju veidi un to specifiskās kategorijas ir attēlotas pārskatāmā tabulā (skat. 4.1. tabula).

4.1. tabula.

Datu kopas aprakstīšanas rezultātā iegūtās kategorijas

Kategorijas veids	Kategorijas nosaukums	Kategoriju aprakstošie objekti
Atrašanās vieta	Mežs	<ul style="list-style-type: none"> • Mežs
	Zālājs	<ul style="list-style-type: none"> • Zāle
	Galvenā pasākuma notikuma vieta	<ul style="list-style-type: none"> • Pludmales karogi (flag banner), • Nožogojuma banneri, • Āras tentis
	Stāvvietas zona	<ul style="list-style-type: none"> • Automašīnas
	Apbalvošanas zona	<ul style="list-style-type: none"> • Pjēdestāls, • Bannera siena
Cilvēku skaits jeb cilvēku grupas lielums	Viens cilvēks	<ul style="list-style-type: none"> • Cilvēki jeb personas
	Divi cilvēki	
	Maza cilvēku grupa	
	Liela cilvēku grupa	
	Cilvēku pūlis	
Cilvēku novietojums pret kameru	No priekšas	<ul style="list-style-type: none"> • Sejas
	Ar muguru	

Tabulā 4.1. minētās kategorijas veido patiešām sarežģītu klasificēšanas uzdevumu. Datoram ir jāiemācās noteikt 10 dažādus objektus, lai veiktu prognozes, balstoties uz to kombinācijām un savstarpējo saistību. Kaut arī cilvēkam šis uzdevums nesagādātu lielas grūtības, datoram šo klasificēšanas uzdevumu nebūs tik pat viegli atrisināt, jo ir daudz objektu un kategoriju, kuras jāņem vērā, lai precīzi noteiktu fotogrāfiju aprakstošo kategoriju kombināciju, tādējādi identificējot fotogrāfijā attēloto.

Doto kolekciju varētu arī klasificēt vēl konkrētāk, piemēram, nosakot cilvēku pozas, kuras apzīmētu cilvēku veiktās aktivitātes, piemēram, skriešana, stāvēšana, gulēšana uz zemes. Taču bakalaura darba ietvaros tika izmantotas tabulā 4.1. minētās kategorijas, kuras ļauj klasificēt fotogrāfijas, balstoties uz fotogrāfiju kolekcijā attēlotajām atrašanās vietām, cilvēku un seju skaitu.

4.3. Izvēlētais ietvars klasifikatora izveidei (Mask R-CNN)

4.3.1. Izvēlētais attēlu atpazīšanas uzdevums

Lai izveidotu klasifikatoru izvēlētai fotogrāfiju kolekcijai, sākumā jāsaprot, kādi objekti un to savstarpējā saistība var palīdzēt identificēt konkrētu pasākuma notikumu. Tabulā 4.1. ir minēti 10 objekti, kurus ir nepieciešams iemācīties atrast fotogrāfijās, lai būtu iespējams prognozēt, kāds notikums ir attēlots fotogrāfijā.

Lai noteiktu atsevišķus objektus fotogrāfijās ir jāizmanto tāds attēlu atpazīšanas uzdevums kā objektu noteikšana (angliski: object detection). Dotais uzdevums atrod objektus, klasificē tos un iezīmē ierobežojošās kastes, kuras palīdz lokalizēt objekta atrašanās vietu attēlā. Šo uzdevumu var veikt tādi algoritmi un ietvari kā, piemēram, R-CNN, Faster R-CNN, YOLO, u.c. Parastais R-CNN ir objektu noteikšanas pirmsākums, taču mūsdienu sarežģītās klasificēšanas problēmas atrisināt ar šo metodi nebūtu jēdzīgi, jo šis algoritms ir pārāk lēns un ir pieejami daudz ātrāki objektu noteikšanas ietvari un algoritmi, kā, piemēram, nodaļā 3. aprakstītie Faster R-CNN un YOLO. Mask R-CNN spēj veikt to pašu, ko Faster R-CNN tikai papildus tas veic arī attēlu segmentēšanas uzdevumu, kas tomēr aizņem nedaudz vairāk laika nekā Faster R-CNN, tāpēc varētu domāt, ka efektīvāk būtu izmantot Faster R-CNN dotās fotogrāfiju kolekcijas objektu noteikšanai. Taču, tā kā iecerētajam klasifikatoram būtu jāmeklē noteikt cilvēku atrašanās vietu fotogrāfijā, tikai ar objektu noteikšanu nepietiek. Nesaskaitāmie objekti kā mežs un zāle gandrīz vienmēr tiek uztverti kā fons, kuram priekšā ir saskaitāmie objekti, kā, piemēram, cilvēki vai automašīnas. Objektu noteikšanas modeļiem treniņdati nav tikai fotogrāfijas un to aprakstošās klases kā tas ir parastajā fotogrāfiju klasificēšanā; objektu noteikšanas modelim trenēšanā ir jāparāda arī, kur konkrētās klases objekts atrodas, tātad kopā ar treniņdatu fotogrāfijām jāpadod arī fails ar visiem attēlotajiem objektiem, objektu klasēm un ierobežojošo kastu koordinātēm. Parasto objektu noteikšanas algoritmu vai ietvaru lietot nebūtu korekti, jo, pieņemsim, ka ir attēlota cilvēku grupa mežā. Cilvēki ir priekšplānā, tāpēc tie nedaudz aizsedz fonu, kas nozīmē, ja tiek iezīmēta meža ierobežojošā kaste, tad cilvēki arī būs tajā iekļauti. Kaut arī CNN spēj pats iemācīties izveidot labākus iezīmju detektorus, labāk būtu datoram uzreiz norādīt, kur tieši atrodas mežs, kas padarītu meža noteikšanu, iespējams, precīzāku un atvieglotu trenēšanas procesu. Tāpēc šajā gadījumā, lai korekti atdalītu objektus no fona, ņemot vērā, ka fons arī ir jānosaka un jāklasificē, papildus objektu noteikšanai ir jāizmanto attēlu segmentēšana.

4.3.2. Izvēlētais segmentēšanas veids un attēlu segmentēšanas ietvars

Kā tika aprakstīts sadaļā 2.2., pastāv 3 attēlu segmentēšanas veidi: semantiskā, instanču un panoptiskā segmentēšana.

Semantiskā segmentēšana pati par sevi nav pielietojama definētā uzdevuma risināšanai, jo ar semantisko segmentēšanu nevar noteikt atrasto objektu skaitu vai individuālo objektu atrašanās vietu – visi vienai klasei piederošie objekti tiek iezīmēti vienā segmentēšanas maskā.

Instanču segmentēšana, savukārt, spēj noteikt katru atsevišķu vienai klasei piederošu instanci, tāpēc tā varētu palīdzēt atrisināt darbā definēto klasificēšanas problēmu. Viens no populārākajiem instanču segmentēšanas ietvariem ir Mask R-CNN, kuram ir pieejama Matterport implementācija; tā ir izveidota programmēšanas valodā Python un balstās uz visizplatītākajām programmēšanas valodas Python mašīnmācīšanās bibliotēkām – Tensorflow un Keras. Pieminētā Mask R-CNN implementācija piedāvā iespēju izmantot uz COCO datukopas iepriekš apmācītu (angliski: pretrained) modeli, kuru var izmantot, lai trenētu instanču segmentēšanas modeli uz dažādu citu objektu noteikšanu un segmentēšanu. Mask R-CNN tika izvēlēts kā iespējamais ietvars definētās klasificēšanas problēmas risināšanai.

Panoptiskā segmentēšana patiesībā būtu labākais attēlu segmentēšanas veids definētās klasificēšanas problēmas risināšanai, jo tā apvieno gan semantisko, gan instanču segmentēšanu: ar semantiskās segmentēšanas palīdzību var noteikt attēloto objektu atrašanās vietu un atdalīt dažādus nesaskaitāmos objektus jeb fona elementus, bet ar instanču segmentēšanas palīdzību var noteikt attēlā esošos objektus, to skaitu, utt. “Facebook AI Researcher’s” ir izveidojuši Detectron2 programmēšanas valodas Python bibliotēku [34], kura piedāvā dažādus objektu noteikšanas un attēlu segmentēšanas algoritmus. Viena no Detectron2 piedāvātajām iespējām ir panoptiskā segmentēšana. Panoptiskās segmentēšanas modelis ir trenēts uz COCO datukopas, kurš māc darboties ar 80 saskaitāmo objektu klasēm (piemēram, cilvēks, automašīna, ābols, u.c.) un ar 91 nesaskaitāmo objektu klasēm, tai skaitā ar fona elementiem (piemēram, zāle, koki, debesis, u.c.). Vairākas fotogrāfijas no izvēlētās kolekcijas tika padotas Detectron2 panoptiskās segmentēšanas modelim un tika noskaidrots, ka tas patiešām spēj segmentēt gandrīz visu attēlā redzamo: gan zāli, gan veģetāciju, gan cilvēkus un mašīnas. Protams, šis risinājums bez papildus trenēšanas neatrisina definēto klasificēšanas problēmu, jo tas nav uztrenēts uz visām tabulā 4.1. definētajām klasēm jeb kategorijām (piemēram, tas nespēj noteikt pjedestālus, cilvēku sejas un konkrēti mežus; tas māc atrast un segmentēt veģetāciju, bet veģetācija ne vienmēr nozīmē, ka objekti atrodas mežā (veģetācija var būt arī redzama fonā, tāpēc salīdzināt veģetācijas un zāles laukumus, lai noteiktu, kas aizņem lielāko attēla daļu, nepalīdzēs korekti prognozēt atrašanās vietu. Panoptiskais segmentēšanas modelis spēj noteikt bannerus jeb vietas, kurās ir kāda reklāma vai sponsori, bet ne kā objektus, piemēram, karogbannerus). Tātad dotais modelis būtu vēl jāuztrenē uz vairākām klasēm, taču tā kā panoptiskā segmentēšana parādījās salīdzinoši nesen, ir ļoti maz pieejamās informācijas par to,

kā panoptisko segmentēšanas modeli trenēt uz citām klasēm. Šī iemesla dēļ sākumā tika izvēlēts Mask-RCNN ietvars, lai izmēģinātu to darbībā un noteiktu, vai tas ir spējīgs atrisināt definēto klasificēšanas problēmu.

4.3.3. Mask R-CNN implementācijas uzstādīšanas process

Praktiskās daļas veikšanai tika izvēlēts Mask R-CNN instanču segmentēšanas ietvars, kuram platformā GitHub ir pieejama implementācija uzrakstīta programmēšanas valodā Python [35]. Tā ir bāzēta uz tādām mašīnmācīšanās bibliotēkām kā Tensorflow un Keras. Mask R-CNN modelis atrod un klasificē objektus, ģenerē ierobežojošās kastes un segmentēšanas maskas katrai objekta instancei attēlā.

Matterport Mask R-CNN implementācija pirmo reizi tika publicēta 2017. gadā un pēdējo reizi tā tika atjaunota 2018. gadā. Tā kā ir pagājuši jau 4 gadi, tad vairākas implementācijā izmantotās bibliotēkas tika atjauninātas, kas rada vairākas versiju saderību problēmas. Implementācijā ir fails ar nosaukumu “requirements.txt”, kurā ir nepieciešamās bibliotēkas Mask R-CNN implementācijas izmantošanai, taču tikai divām bibliotēkām ir pierakstītas versijas: tensorflow ($\geq 1.3.0$) un keras ($= 2.0.8$). Pārējām 10 bibliotēkām nav pierakstītas versijas, tāpēc ar komandu “pip install -r requirements.txt” tiek automātiski instalētas pēdējās bibliotēku jeb pakotņu versijas. Tas rada neskaitāma daudzuma kļūdu, tāpēc sākumā bija jānoskaidro, kādas versijas ir nepieciešamas, lai dotā Mask R-CNN implementācijas darbotos korekti. Diezgan daudz laika tika patērēts, darbinot Mask R-CNN, lasot kļūdas un meklējot kļūdu paziņojumu iemeslus internetā. Galu galā izdevās atrast visas nepieciešamās bibliotēku versijas, lai Mask R-CNN implementācija darbotos korekti un bez kļūdām; implementāciju izdevās palaist ar pēdējo Tensorflow 1.x versiju (1.15.0).

Lai Mask R-CNN implementācija darbotos korekti, “requirements.txt” failā 2 bibliotēkām tika pierakstītas specifiskas versijas: scikit-image (0.16.2) un h5py (2.10.0), kā arī tika izmainītas 2 sākumā definētās tensorflow un keras versijas: tensorflow (1.15.0) un keras (2.3.0). Pārējās bibliotēkas izmantoja pēdējās pieejamās versijas. Ar šo versiju specifikāciju izdevās palaist visus Mask R-CNN implementācijas piemēru kodus, lai pārlicinātos, ka viss darbojas korekti, un vēlāk arī tika uztrenēts Mask R-CNN instanču segmentēšanas modelis uz izvēlētās fotogrāfiju kolekcijas definēto kategoriju apzīmējošiem objektiem, kuri ir definēti tabulā 4.1.

Tā kā sākumā tika veikti nelieli testi uz personīgā datora, lai noskaidrotu, vai Mask R-CNN būs spējīgs noteikt nesaskaitāmus objektus attēlā, kā mežs un zāle, tensorflow (1.15.0)

pakotne, kura veic CPU skaitļošanu jeb skaitļošanu, izmantojot centrālo procesoru, tika aizvietota ar tensorflow-gpu (1.15.0), lai nelielu treniņu veikšana neaizņemtu daudz laika. GPU skaitļošana ir daudz ātrāka nekā CPU skaitļošana. Lai dators spētu veikt GPU skaitļošanu, ir nepieciešama NVIDIA grafikas karte (uz personīgā datora, kas ir klēpjdatore, ir uzstādīta Nvidia GTX1050 4GB), Visual Studio, CUDA rīku komplekts (angliski: toolkit) un cuDNN (“Cuda Deep Neural Network”) bibliotēka. Lai tensorflow pakotne spētu izmantot grafikas karti skaitļošanai, Visual Studio, CUDA, cuDNN un tensorflow versijām ir jābūt saderīgām. Izmantojot Nvidia un Tensorflow oficiālās versiju saderību tabulas, tika noskaidrots, ka, lai tensorflow-gpu (1.15.0) spētu izmantot GPU skaitļošanai, ir jābūt sekojošām komponentu versijām:

- Visual Studio (core only) – Visual Studio 2017;
- CUDA toolkit – 10.0;
- cuDNN – 7.6.5;
- Nvidia Graphics driver – 457.63.

Ar augstāk minēto specifikāciju izdevās iespējot GPU skaitļošanu, taču 1. trenēšanas procesā tika noskaidrots, ka nezināmu iemeslu dēļ klēpjdatore, kuriem ir integrētā un papildus grafikas karte, pēc pirmā trenēšanas cikla (angliski: epoch) netiek pāriets uz nākamo, bet process bezgalīgi turpinās (lai beigtu procesu, manuāli jābeidz process, izmantojot uzdevumu pārvaldnieku). Lai šo problēmu atrisinātu, Windows 10/11 grafikas iestatījumos jāieslēdz slēdzis ar nosaukumu “Hardware-accelerated GPU scheduling”.

Veicot visas augstāk aprakstītās darbības, izdevās sekmīgi iespējot instanču segmentēšanas modeļa trenēšanu, izmantojot GPU skaitļošanu.

Papildus oficiālajai Matterport Mask R-CNN implementācijai no platformas GitHub tika izmantots vēl viens Python fails [36], kurš apkopo vairākas Mask R-CNN implementācijas funkcionalitātes ērtāk lietojamās funkcijās, piemēram, ir definētas tādas funkcijas kā “train_head()” vai “load_training_model()”; kā arī dotais fails piedāvā funkciju datukopas un trenēšanā izmantoto anotāciju ielasīšanai. Anotāciju ielasīšanas funkcionalitāte ir izveidota tā, lai tiktu atbalstīta “makesense.ai” anotēšanas rīka veidotais anotāciju fails. Anotēšanas rīks “makesense.ai” ir aprakstīts sadaļā 4.4., kurš arī tika izmantots treniņkopas fotogrāfiju anotēšanai.

4.3.4. Iepriekš uztrenēts COCO modelis

Matterport Mask R-CNN implementācija piedāvā iespēju izmantot uz COCO datu kopas iepriekš apmācītu modeli. COCO datu kopa satur 80 saskaitāmu objektu klases, tātad dotajam

modelim jau ir vispārīga saprašana par vairākiem objektiem un to iezīmēm. Iepriekš apmācītus modeļus var izmantot, lai pielāgotu tos un izmantotu to zināšanas citu, līdzīgu uzdevumu risināšanā – šis process saucās TL [37].

TL ideja ir sekojoša: izmantot iepriekš iemācītas iezīmes uz kādas vienas problēmas, un izmantot tās citai, līdzīgai problēmai. Piemēram, iemācītās iezīmes no automašīnu noteikšanas problēmas var būt izpalīdzīgas, lai noteiktu smagās automašīnas attēlā. TL pārsvarā izmanto tādus gadījumus, kad datukopa, uz kuras ir doma trenēt mašīnmācīšanās modeli, ir pārāk maza, lai uztrenētu labu modeli no “nulles” jeb no nekā. TL notiek vairākos soļos: sākumā tiek paņemti neironu tīkla slāņi no iepriekš apmācīta modeļa, tad šie slāņi tiek iesaldēti, lai jau apgūtās zināšanas netiktu iznīcinātas trenēšanas procesā, tālāk tiek izveidoti jauni apmācāmi slāņi pa virsu iesaldētiem slāņiem un beigās jaunie apmācāmie slāņi tiek trenēti uz jaunas datu kopas. Ja TL tehnika nedod vēlamo rezultātu, tad pēdējais solis ir FT, kas apzīmē procesu, kurā visi iepriekš uztrenēta modeļa slāņi vai tā daļas tiek atsaldētas un trenētas uz jauniem datiem. FT gadījumā tiek izmantots ļoti mazs LR, lai netiktu pārāk stipri izmainītas iepriekš apgūtās zināšanas. FT pakāpeniski var palīdzēt uzlabot modeli, adaptējot iepriekš apgūtās zināšanas jauniem datiem.

4.4. Treniņdatu sagatavošanas process

Tā kā Mask R-CNN implementācija piedāvā iespēju izmantot uz COCO datu kopas jau iepriekš apmācītu modeli, kurš spēj atpazīt 80 dažādus objektus, t.sk. cilvēkus un automašīnas, nav nepieciešama milzīga treniņkopa.

Izvēlētā fotogrāfiju kolekcija satur 1172 fotogrāfijas. No tām tika izvēlētas 96 fotogrāfijas instanču segmentēšanas modeļa trenēšanai. Patiesībā 96 fotogrāfijas ir ļoti maza datukopa trenēšanai, ņemot vērā to, ka modelis ir jātrenē uz 8 jaunām klasēm, taču, lai uztrenētu attēlu segmentēšanas uzdevuma modeli, katram treniņkopas attēlam ir manuāli jāiezīmē visu vēlamo attēloto objektu segmentācijas maskas. Tas ir patiešām ļoti smalks un laikietilpīgs uzdevums. Tika mēģināts atlasīt tādas fotogrāfijas, kuras maksimāli labi attēlo vēlamos objektus, lai modelis labi iemācītos šo objektu iezīmes.

Tā kā definētās klasificēšanas problēmas risināšanai tika izvēlēts Mask R-CNN kā instanču segmentēšanas ietvars, tad nepietiek modelim padot tikai fotogrāfijas un ierobežojošo kastu koordinātes. Papildus ir jāpadod segmentēšanas maska jeb anotācijas, kuras ir reprezentētas ar poligoniem jeb daudzstūriem.

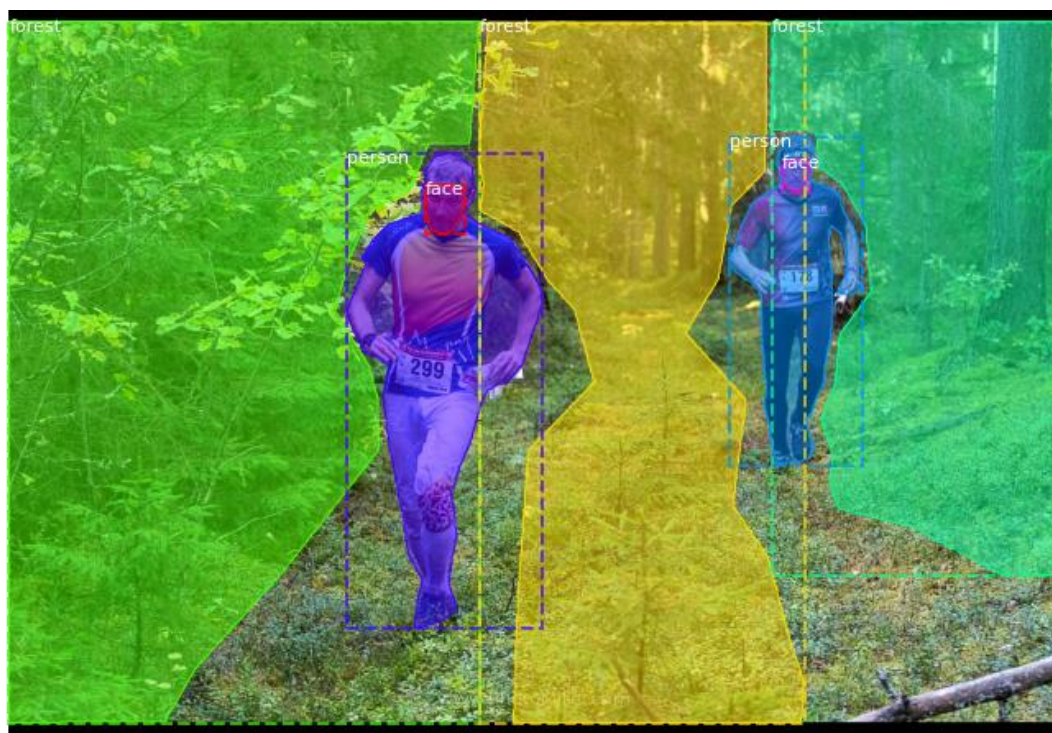
Eksistē daudzi dažādi datu anotēšanas rīki, kuri spēj iezīmēt gan ierobežojošās kaste, gan poligonus, gan punktus u.tml. Kā tika aprakstīts sadaļas 4.3.3. beigās, papildus Matterport Mask

R-CNN implementācijai tika izmantots fails ar funkcijām, kuras apkopo vairākas Mask R-CNN implementācijas funkcionalitātes ērtākai lietošanai. Šajā papildus failā ir piedāvāta arī iespēja ielasīt datukopu un datukopas anotācijas (angliski: annotations) ar nosacījumu, ka anotāciju fails tika veidots ar rīku “makesense.ai”. Minētais rīks tika izmantots treniņdatu anotēšanai.

“makesense.ai” [38] ir datu anotēšanas rīks, kurš ir pieejams gan platformā Github lokālai instalēšanai, gan kā tīmekļa lietotne, kas neprasa nekādas papildus darbības jeb instalēšanu – pietiek tikai atvērt pārlūkprogrammu. “makesense.ai” piedāvā vairāku anotēšanas veidu atbalstu: taisnstūri, līnijas, punkti un poligoni. Kā arī tiek atbalstīti vairāki failu izvades formāti, t.sk. YOLO, VGG JSON, CSV un COCO JSON. Treniņkopas anotēšanas procesā tika izmantota “makesense.ai” tīmekļa lietotnes versija. Tai ir ļoti patīkams dizains un patiešām ērta saskarne.

Kaut arī “makesense.ai” rīks kopumā ir labs, treniņkopas anotēšanas procesā tika noteiktas divas “makesense.ai” rīka problēmas: viena no ērtas lietojamības viedokļa un otra no funkcionalitātes viedokļa. Lietojamības problēma ir sekojoša: veicot objekta iezīmēšanu ar poligona tipu, nav iespējams izdzēst pēdējo uzlikto punktu, ja tas tika uzlikts nepareizajā vietā – vai nu ar “Escape” pogu var pilnībā izdzēst poligonu un sākt no sākuma, vai arī var pabeigt iezīmēt poligonu (savienot beigu punktu un sākuma punktu, izveidojot vienu veselu poligonu) un tikai tad izmainīt punkta atrašanās vietu (redīgēt poligonu). Šīs problēmas atrisināšana varētu ietaupīt vairāk laika, bet tā tieši neietekmē anotāciju kvalitāti un tā pilnību. Funkcionalitātes problēma, savukārt, gan ietekmē – fotogrāfijā var gadīties, ka objekts #1 daļēji aizsedz objektu #2 tā, ka attēlā objekts #2 ir sadalīts uz pusēm jeb objektam #2 šajā attēlā būtu jāiezīmē 2 segmentēšanas maskas. COCO datu anotēšanas formāts dod iespēju vienam objektam piešķirt sarakstu ar segmentācijām, taču “makesense.ai” rīks katru jaunu poligonu uzskata kā atsevišķu objektu jeb instanci (nav iespējams definēt, kuram konkrētam objektam jeb instancei pieder iezīmētais poligons). Vienīgais veids, kā atrisināt problēmu, ir eksportēt uzģenerēto anotāciju failu un manuāli pievienot papildus segmentēšanas konkrētiem objektiem, bet tas būtu ļoti sarežģīti un aizņemtu daudz laika. Izveidotajā treniņkopā nebija daudz tādu attēlu, kur būtu jāiezīmē vairākas segmentācijas vienam objektam, tāpēc šī problēma stipri neietekmēja modeļa trenēšanas procesu un modeļa precizitāti, taču tā nedaudz ietekmēja instanču segmentēšanas modeļa segmentācijas masku iezīmēšanu.

4.4.1. Anotētie treniņdatu piemēri



4.8. att. Anotētā fotogrāfija – divi sacensību dalībnieki mežā

Attēlā 4.8. tika iezīmētas 7 segmentēšanas maskas: 2 cilvēku maskas, 2 seju maskas un 3 meža maskas. Kaut arī Mask R-CNN iepriekš apmācītais modelis māc atpazīt cilvēkus, veicot jauna objektu noteikšanas modeļa izstrādi (trenējot modeli uz jaunām klasēm), tiek noņemts klasificēšanas slānis jeb slānis, kurš satur 80 COCO datu kopas klases, un klasificēšanas slānis tiek aizvietots ar jaunu klasificēšanas slāni, kurš satur jaunās definētās klases, t.sk. arī tās, kuras ir vēlme paturēt no COCO datukopas klasēm (šajā gadījumā tās ir klases “cilvēks” un “automašīna”); protams, jāiezīmē arī cilvēki un automašīnas, jo iepriekš apgūtām zināšanām jāsavienojas ar jauno klasificēšanas slāni). Bet tā kā iepriekš apmācītais modelis jau māc atpazīt cilvēkus, jaunajam modelim nebūs lielas grūtības atkal iemācīties klasificēt cilvēkus, jo visos iesaldētajos slāņos jau ir saglabāta informācija par cilvēku iezīmēm; varbūt ar mazu datu kopu jaunais modelis nespēs atpazīt cilvēkus tik pat precīzi kā uz COCO datukopas iepriekš apmācīts modelis, taču tas spēs atpazīt cilvēkus diezgan labi. Cilvēku sejas palīdz noteikt, cik cilvēku ir vērsti pret kameru ar sejām jeb fotografēti no priekšas, tādējādi fotogrāfijā attēlotais notikums tiek detalizētāk aprakstīts. Ar meža iezīmēšanu ir daudz sarežģītāk. Pirmkārt, meža iezīmes atšķiras, mainot skatu punktu. Atrodoties ārpus meža, ir redzamas vairāk koku lapas vai skujuas, stumbri ir salīdzinoši tievi, jo tie atrodas tālumā, sumbru tekstūra nav labi saskatāma, kā arī pārsvarā ir redzami tikai priekšplānā esoši koki (meža robeža). Atrodoties iekšā mežā, koku stumbri ir diezgan tuvu, tāpēc ir redzama gan koku mizas tekstūra (koku mizas iezīmes), gan arī ir redzami koki tuvumā un tālumā; balstoties uz šīm iezīmēm, var noteikt, vai ir redzams mežs no “iekšpuses” vai no ārpuses. Tas, ka fotogrāfijā ir redzams mežs kā fons, nenožīmē, ka

attēlojamie objekti atrodas mežā (tā ir tikai papildus informācija par apkārtni). Lai noteiktu, ka objekti atrodas iekšā mežā, objektu noteikšanas modelim jāiemācās specifiskas meža iezīmes, kuras var saredzēt tikai atrodoties mežā. Attēlā 4.8. tika iezīmētas vietas, kur ir redzamas priedes, priežu stumbri, dažādu citu koku lapas tuvumā; šīs iezīmes liecina par to, ka fotogrāfija attēlo mežu no iekšpuses. Var pamanīt, ka netika iezīmēta zeme, uz kura stāv cilvēki. Mežos uz zemes gandrīz vienmēr atrodas sūnas, tāpēc nebūtu korekti iezīmēt attēlā 4.8. sūnas zem kategorijas jeb klases “zāle”. Sākumā bija doma iezīmēt arī sūnas kā meža segmentēšanas maskas daļu, bet tas netika izdarīts, jo fotogrāfijās zāle un sūnas izskatās ļoti līdzīgi no attāluma, tāpēc, iespējams, modelim būtu grūtāk noteikt zāles laukumus (zāle varētu tikt klasificēta kā mežs).

Mask R-CNN nespēj veikt semantisko segmentēšanu, tāpēc vienīgais veids, kā iezīmēt mežu vai zāli, ir izmantojot poligonu iezīmēšanu. Instanču segmentēšanas modelis, iespējams, atradīs vairākas meža vai zāles instances, taču, lai noteiktu fotogrāfijā atrašanās vietu, pietiek tikai atrast vismaz vienu meža vai zāles instanci (pat, ja tiks atrastas vairākas meža vai zāles instances, fotogrāfijas notikuma atrašanās vietas kategorija nemainīsies, ja tie ir vienīgie objekti, kuri tika atrasti).



4.9. att. Anotētā fotogrāfija – sacensību dalībnieks un skatītājs galvenajā pasākuma notikuma vietā

Nožogojuma banneri parasti atrodas sacensību maršruta zonā. Izvēlētajā fotogrāfiju kolekcijā tie atrodas pašās beigās jeb pie finiša, tāpēc dotajā kolekcijā nožogojuma banneri apzīmē finiša zonu, taču daudzos citos maratonos banneru nožogojumi vai vienkārši

nožogojumi var atrasties arī sākumā vai pat visa maršruta garumā; tāpēc pieņemt, ka nožogojuma banneri vienmēr apzīmē finiša zonu, nebūtu korekti universāla klasifikatora izveidei (vajadzētu iegūt vairākas finiša un starta pazīmes, ja tādas ir).

Attēlā 4.9. var pamanīt, ka netika iezīmēts mežs, jo dotajā fotogrāfijā mežs neapzīmē atrašanās vietu, bet gan fonu, tāpēc modelim nebūtu jānācās meža iezīmes, kuras piemīt, skatoties uz mežu no tāluma. Varētu definēt 2 meža apakškategorijas – “mežs kā fons” un “mežs no iekšpuses” –, lai varētu iegūt papildus informāciju par apkārtējo vidi, tādējādi iegūstot vēl vairāk informācijas, taču šajā darbā tika izņemts mācīt modeli tikai uz meža iezīmēm no “iekšpuses”, lai noteiktu cilvēku atrašanās vietu fotogrāfijā.



4.10. att. Anotētā fotogrāfija – liela cilvēku grupa galvenajā pasākuma notikuma vietā

Attēlā 4.10. tika iezīmēti gandrīz visi labi saskatāmi cilvēki. Ar šo būtu jāpietiek, lai jaunais modelis varētu saistīt iepriekš iemācītās zināšanas par cilvēkiem, lai spētu atrast un klasificēt cilvēkus. Mežs šajā fotogrāfijā, tāpat kā attēlā 4.9., netika iezīmēts, jo mežs apzīmē fonu.



4.11. att. Anotētā fotogrāfija – divi organizatori ar mašīnu fonā

Attēlā 4.11. var pamanīt ka fonā atrodas viena automašīna, taču ir redzamas 3 segmentēšanas maskas, kuras tiek attēlotas kā 3 dažādas automašīnu instances. Šajā anotācija var vizuāli noprast “makesense.ai” datu anotēšanas rīka funkcionalitātes problēmu.

4.5. Instanču segmentēšanas modeļa trenēšanas process

Kad Mask R-CNN implementācija tika sekmīgi uzstādīta un trenēšanas dati sagatavoti, tika sākts modeļa trenēšanas process. Tas sastāv no datu kopas sadalīšanas treniņkopā un validācijas kopā, trenēšanas hiperparametru noteikšanas un iegūto rezultātu analīzes.

Treniņkopa (angliski: training set), validācijas kopa (angliski: validation set) un testa kopa (angliski: test set) [39] ir sākotnējās izvēlētās kopas sadalījums apakškopās; katrai no šīm apakškopām ir sava nozīme modeļa mācīšanās procesā. Treniņkopa ir fotogrāfiju kopa, uz kuras modelis mācās noteikt nepieciešamās objektu iezīmes, lai turpmāk veiktu konkrēto uzdevumu, balstoties uz zināšanām. Treniņkopa tiek padota modelim vairākas reizes trenēšanās laikā, lai tas uzlabotu savas zināšanas un precizitāti. Validācijas kopa ir fotogrāfiju kopa, kura tiek izmantota katra mācīšanās cikla (angliski: epoch) beigās, lai noteiktu, kā modelis veic prognozēšanu uz fotogrāfijām, kuras tas nav redzējis. Balstoties uz tādiem validēšanas parametriem kā, piemēram, zaudējuma funkcijas vērtību vai validācijas precizitāti, var noteikt, vai modelis pārmācās iezīmes no treniņkopas jeb pārmērīgi pielāgojas (angliski: overfit)

treniņkopā esošiem attēliem – tas nozīmē, ka modelis spēs ļoti labi, piemēram, klasificēt treniņkopas fotogrāfijas, bet citas, iepriekš neredzētas fotogrāfijas (piemēram, no validācijas vai testa kopas) tas klasificēs ar ļoti zemu precizitāti. Validācijas kopa var palīdzēt noteikt optimālās trenēšanas procesa hiperparametru vērtības, lai uzlabotu modeļa precizitāti. Testa kopa, savukārt, tiek izmantota tad, kad modelis jau ir uztrenēts un ir jāveic tā precizitātes novērtēšana jeb spēja pildīt uzdoto uzdevumu.

Hiperparametri [40] ir parametri, kuru vērtības kontrolē mācīšanās procesu un nosaka trenējamā modeļa parametru vērtības, kuras mācīšanās algoritms iemācās. Modeļa parametri un hiperparametri ir divi dažādi parametru tipi. Modeļa parametri ir mainīgie, kuri atrodas iekšā pašā modelī un kuri tiek mainīti mācīšanās procesā; izmantotais mācīšanās algoritms mēģina iemācīties “savienot” iezīmes ar klasēm jeb kategorijām. Hiperparametri, savukārt, ir modeļa trenēšanas procesa parametri – tie nav daļa no modeļa. Hiperparametri būtiski ietekmē rezultējošā modeļa precizitāti jeb spēju pildīt konkrētu uzdevumu. Neeksistē tādas hiperparametru vērtības (optimālākās vērtības), kuras varētu izmantot jebkura uzdevuma risināšanai. Katram uzdevumam optimālākās hiperparametru vērtības ir individuālas, tāpēc vienīgais veids, kā noteikt šos parametrus, ir veicot vairākus eksperimentus, mainot hiperparametrus, lai analizētu rezultātus un noteiktu, kādas hiperparametru vērtības veicina visprecīzākā modeļa izveidi. Protams, hiperparametru vērtību kombinācijas var būt ļoti daudz, tāpēc optimālāko hiperparametru noteikšana ir laikietilpīgs process.

Instanču segmentēšanas modeļa trenēšana bakalaura darba ietvaros tika veikta vairākus desmitus reižu ar dažādām hiperparametru vērtībām un dažām treniņkopu un validācijas kopu sadalīšanas attiecībām.

4.5.1. Trenēšana uz personīgā datora un Google Colab vidē

Veicot pirmos instanču segmentēšanas modeļa trenēšanas mēģinājumus uz visām 10 definētajā klasēm, tika secināts, ka uz personīgā datora nepietiek resursu, lai trenētu lielus neironu tīklus. Mask R-CNN implementācija ir bāzēts uz ResNet101 neironu tīkla arhitektūras, kas ir neironu tīkls ar 101 slāni – tas ir ļoti dziļš neironu tīkls. Uz personīgā datora izdevās tikai uztrenēt iepriekš apmācīta modeļa galvas (angliski: head) slāņus. Mēģinot uztrenēt vairāk slāņus (piemēram, sākot no ResNet101 4. konvolūcijas līmeņa līdz beigām, ieskaitot galvas slāņus), tika iegūti OOM (“Out Of Memory”) kļūdas paziņojumi, kas ziņo par to, ka tika pārsniegta GPU atmiņas limits. 4GB grafikas kartes atmiņas nepietiek, lai trenētu ar TL tehnikas palīdzību iepriekš apmācītu modeli, kas izmanto ļoti dziļus neironu tīklus. Uzņēmums Google piedāvā tādu platformu kā Google Colab, kas nodrošina iespēju jebkuram rakstīt un palaist patvaļīgu programmēšanas valodas Python kodu, izmantojot tīmekļa pārlūkprogrammu, kā arī Google Colab ir ļoti labi piemērots mašīnmācīšanai un datu analīzei. Google Colab

piedāvā vairākus abonementu veidus, kur viens no tiem ir “Free plan” jeb bezmaksas plāns. Bezmaksas plāns piedāvā GPU skaitļošanas iespēju ar 12GB grafikas kartes atmiņu, taču pēc ilgas lietošanas tiek piešķirts pārtraukums uz nenoteiktu laiku, kas ir vienīgais ierobežojums, ņemot vērā, ka ir iespējams izmantot GPU ar patiešām lielu video atmiņas apjomu.

4.5.2. Treniņdatu sadalīšana apakškopās un hiperparametru noteikšana

Pēc noklusējuma uzstādītā Mask R-CNN implementācija kopā ar papildus failu ar funkcijām sadala anotēto datukopu no 96 fotogrāfijām treniņkopā un validācijas kopā ar attiecību 90/10. Parasti 90/10 attiecību izmanto gadījumos, kad trenēšanas datu kopa ir liela [39]. Tā kā izveidotā anotētā datu kopa ir ļoti maza, tad datu kopa trenēšanai tika sadalīta treniņkopā un validācijas kopā ar attiecību 80/20, lai validācijas kopa vairāk reprezentētu izvēlēto fotogrāfiju kolekciju un lai būtu pietiekami daudz treniņdatu modeļa trenēšanai. Tika testētas arī attiecības 90/10 un 70/30, bet 80/20 izrādījās vispiemērotākā attiecības izveidotajai un anotētajai trenēšanas datu kopai. Attiecība 90/10 nederēja modeļa instanču segmentēšanas spēju novērtēšanai, jo bija par maz attēlu validācijas kopā (nevarēja noteikt, kāda hiperparametru kombinācija varētu būt optimālākā), savukārt, attiecība 70/30 nederēja, jo treniņdatu kopa tādā gadījumā bija pavisam maza un modelis diezgan slikti veica nepieciešamo objektu noteikšanu, kā arī bija grūti novērtēt uzlabojumus, mainot hiperparametrus. Sadalot anotēto datu kopu ar attiecību 80/20 tika iegūta treniņkopa, kura satur 76 fotogrāfijas un validācijas kopa, kur satur 20 fotogrāfijas.

Instanču segmentēšanas modeļa trenēšanas laikā tika mainīti tādi hiperparametri kā ciklu (angliski: epochs) skaits, SPE, BS un LR. Pārējie hiperparametri palika tādi paši kā Mask R-CNN implementācijā pēc noklusējuma.

Viens no pirmajiem, labajiem instanču segmentēšanas modeļiem tika uztrenēts divos posmos ar tabulā 4.2. minētajiem hiperparametriem.

(Treniņkopas izmērs – 76; Validācijas kopas izmērs – 20)

Trenēšanas posmi:

- 1.posms – Galvas jeb pēdējo modeļa slāņu trenēšana.
- 2. posms – Visu slāņu trenēšana/pielāgošana (FT).

Viens no vairākiem trenēšanas procesiem, kurš palīdzēja uztrenēt salīdzinoši labu instanču segmentēšanas modeli

Posms	Hiperparametri			
	BS	Cikli	SPE	LR
1.posms	1	10	200	0.001
2.posms	1	10	200	0.0001

Modelis, kurš tika trenēts 2 posmos ar tabulā 4.2. minētajiem hiperparametriem, spēja labi atpazīt vairākus, taču ne visus cilvēkus priekšplānā, dažas automašīnas, kaut cik labi atrada mežu un zāli, taču segmentācijas maskas bija mazas un neprecīzas. Sejas atrada salīdzinoši maz un āras tentis gandrīz vispār neatrada. Izvēlētais soļu skaits vienā ciklā jeb SPE (200) bija izvēlēts tikai mēģinājuma dēļ. Balstoties uz pieejamo informāciju par neironu tīklu trenēšanu, pārsvarā SPE tiek definēts tā, lai vienā ciklā modelis izietu cauri visai treniņkopai vienu reizi [41]. Lai iegūtu SPE, ir jāizdala treniņkopas paraugu (angliski: samples) skaits ar BS. Izmantojot tradicionālo SPE rēķināšanas pieeju, 1.posmā SPE vērtībai būtu jābūt $76/2$ (38) nevis 200. Tā kā Mask R-CNN implementācija neizmanto LR plānotāju (angliski: scheduler) jeb LR hiperparametrs netiek mainīts pēc konkrēta iterāciju vai trenēšanas ciklu skaita, tas nozīmē, ka palielināts SPE ir ekvivalents palielinātam ciklu skaitam, t.i. vienā ciklā modelis iziet cauri treniņkopai vairākas.

Vēl viens trenēšanas mēģinājums bija veikts ar augstāk minētajiem hiperparametriem, vienīgi SPE tika izmanīts no 200 uz 400. Apmācītais modelis veica objektu noteikšanu un segmentēšanu daudz sliktāk nekā modelis, kurš tik trenēts ar SPE 200; modelis sāka pārmācīties treniņkopas attēlos esošās objektu iezīmes.

Nākamajos mēģinājumos tomēr tika izmantota tradicionālā SPE hiperparametra vērtības rēķināšana. Galu galā, balstoties uz instanču segmentēšanas modeļa rezultātiem uz validācijas kopas attēliem, tika secināts, ka tabulā 4.3. minētie dati ir optimālākie noteiktie trenēšanas posmi un to hiperparametri.

(Treniņkopas izmērs – 76; Validācijas kopas izmērs – 20)

Trenēšanas posmi:

- 1.posms – Galvas jeb pēdējo modeļa slāņu trenēšana.
- 2. posms – ResNet101 slāņu trenēšana, sākot no Conv4 uz augšu (FT)

Katra trenēšanas posma optimālākie hiperparametri instanču segmentēšanas modeļa trenēšanai uz 10 definētajām klasēm (kategorijām)

Posms	Hiperparametri			
	BS	Cikli	SPE	LR
1.posms	2	20	38 (76/2)	0.001
2.posms	1	40	76 (76/1)	0.00033

Kopā modelis tika trenēts 60 ciklus. Galvas slāņi tika trenēti ar BS 2, jo tā bija gandrīz maksimālā vērtība (BS 4 jau Google Colab vidē atgriezta OOM kļūdas paziņojumu; tas ir tāpēc, Mask R-CNN modelis ievadē sagaida 1024x1024 attēlus, kas ir patiešām liels attēls). Lielāks BS padara trenēšanas procesu ātrāku.

Tika arī mēģināts turpināt trenēšanu pēc 60 cikliem, veicot visu iepriekš apmācīta modeļa slāņu trenēšanu 20 un 40 ciklus (3.posms), taču iegūtie modeļi izrādījās daudz sliktāki nekā modelis, kurš tika trenēts 2 posmos. Iespējams, tas ir tāpēc, ka iepriekš apgūtās zināšanas un to saistība ar konkrētiem objektiem tika pa daudz izmainīta, tādējādi kaut kādas apgūtās zināšanas tika iznīcinātas.

4.5.3. Trenēšana uz visas anotētās datu kopas (96 attēli)

Tā kā tika iegūti iespējamie optimālie hiperparametri, vēlreiz tika veikts tas pats trenēšanas process, kas veicināja labākā instanču segmentēšanas modeļa precizitāti (skat. 4.3 tabula), balstoties uz validācijas kopas analizēšanu, taču treniņkopa un validācijas kopa tika apvienotas vienā kopā, lai modelim būtu vairāk treniņdatu, uz kuriem trenēties. Tādējādi gala varianta trenēšanas procesa posmi un to hiperparametri ir attēloti tabulā 4.4.

(Treniņkopas izmērs – 96; Validācijas kopas izmērs – 0)

Trenēšanas posmi:

- 1.posms – Galvas jeb pēdējo modeļa slāņu trenēšana.
- 2. posms – ResNet101 slāņu trenēšana, sākot no Conv4 uz augšu (FT)

Pēdējā trenēšanas procesa hiperparametri, izmantojot visus anotētos datus

Posms	Hiperparametri			
	BS	Cikli	SPE	LR
1.posms	2	20	48 (96/2)	0.001
2.posms	1	40	96 (96/1)	0.00033

Kad instanču segmentēšanas modelis tika uztrenēts, izmantojot tabulā 4.4. minētos trenēšanas posmus un hiperparametrus, tika sākta modeļa testēšanas fāze, lai noteiktu modeļa trūkumus un, balstoties uz šiem trūkumiem, izveidotu uztrenētajam modelim maksimāli precīzu klasificēšanas algoritmu.

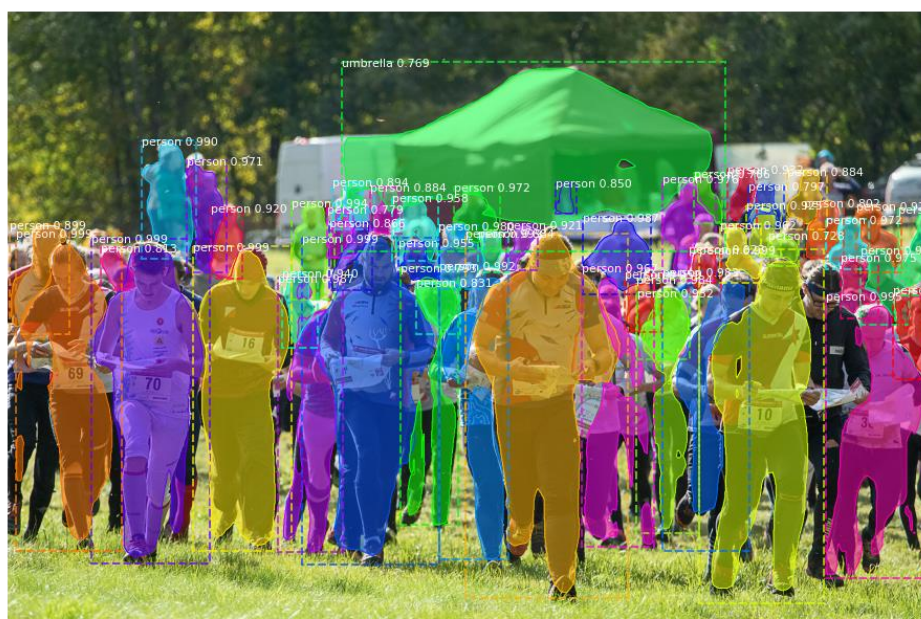
4.6. Uztrenētā instanču segmentēšanas modeļa testēšana un rezultāti

4.6.1. Iepriekšapmācīta modeļa instanču segmentēšanas salīdzināšana ar darba autora uztrenēto modeli

Iepriekšapmācīts modelis trenēts uz COCO datukopas māk noteikt un segmentēt 80 dažādas klases, tai skaitā 2 klases, kuras ir nepieciešamas izvēlētās fotogrāfiju kolekcijas augsta līmeņa klasificēšanai: cilvēks un automašīna.

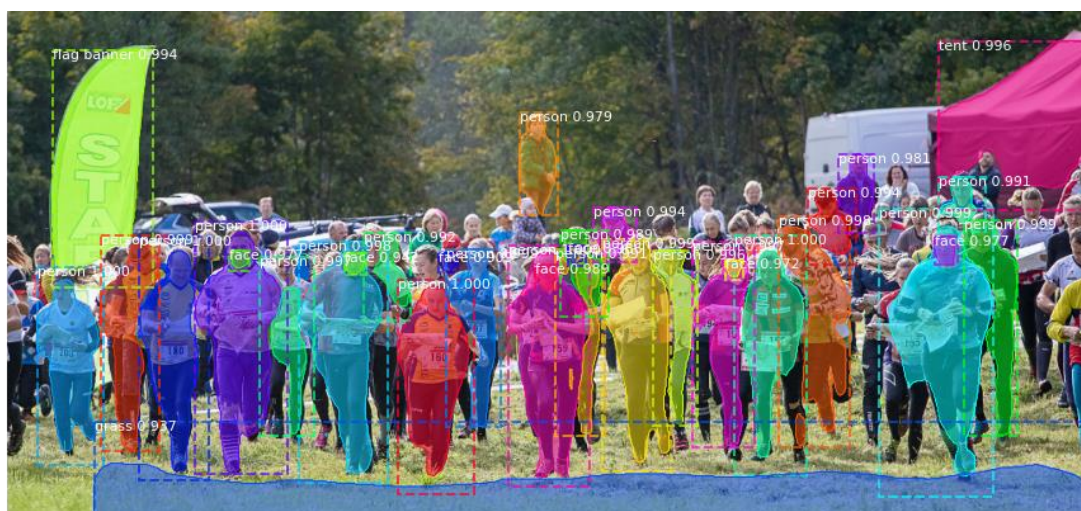
Tā kā veicot TL tiek izveidots jauns klasificēšanas slānis ar jaunajām definētajām klasēm jeb kategorijām, iepriekšapmācīta modeļa zināšanas pazaudē saikni ar izvades slānī esošajām kategorijām, pat ja tajā ir tādas klases, uz kurām iepriekšapmācīts modelis jau tika trenēts (šajā gadījumā tās ir klases “cilvēks” un “automašīna”). Iepriekš apgūtās zināšanas tiek savienotas ar jauno klasificēšanas slāni, taču saitēm, kuras savieno iepriekšapmācītu modeli ar jauno klasificēšanas slāni, tiek piešķirti nejauši svāri, tāpēc darba autora trenētajam modelim atkal ir jābūvējas saprast, kuras iezīmes piemīt klasēm “cilvēks” un “automašīna”, taču, tā kā modelis jau ir apmācīts uz šīm klasēm, tad, iespējams, tas spēs daudz ātrāk (atkal) iemācīties noteikt un klasificēt cilvēkus un automašīnas.

Zemāk ir salīdzināti iepriekšapmācīta modeļa un darba autora uztrenētā modeļa rezultāti uz divām izvēlētās kolekcijas fotogrāfijām, kurās ir attēlots liels cilvēku pūlis.



4.12. att. Iepriekšapmācītā modeļa atrastie objekti attēlā #1

Attēlā 4.14. ir redzams testējamais attēls #2, kurš tika segmentēts ar iepriekšapmācītu modeli. Tāpat kā attēlā 4.12., iepriekšapmācītais modelis atrada un iezīmēja gandrīz visus cilvēkus.



4.15. att. Darba autora uztrenētā modeļa atrastie objekti attēlā #2

Attēlā 4.15. ir redzams testējamais attēls #2, kurš tika segmentēts ar darba autora uztrenēto modeli. Modelis atrada pludmales karogu, tenti, un vairākas sejas, taču dotais modelis, tāpat kā attēlā 4.13., atrada tikai tos cilvēkus, kuri ir attēloti pilnā augumā vai līdz viduklim.

Var secināt, ka viens no darba autora uztrenētā modeļa trūkumiem, ir cilvēku atpazīšana gadījumos, kad tie stāv aiz citiem cilvēkiem tā, ka ir redzama tikai galva vai cilvēks līdz pleciem.

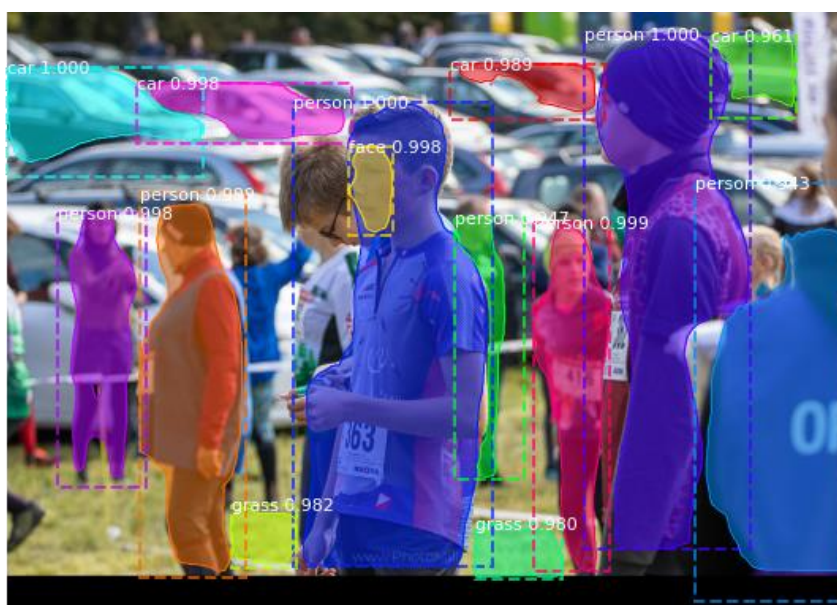
4.6.2. Rezultāti no testa kopas

Izvēlētās fotogrāfijas testa kopa tika izveidota, atņemot no visām 1172 kolekcijas fotogrāfijām treniņkopā esošās 96 fotogrāfijas. Tādējādi iznāk, ka testa kopa sastāv no 1076 fotogrāfijām. Vairāki desmiti fotogrāfiju no testa kopas tika izmantoti gan instanču segmentēšanas modeļa novērtēšanai, gan augsta līmeņa klasifikatora novērtēšanai.



4.16. att. Dalībnieki apbalvošanā

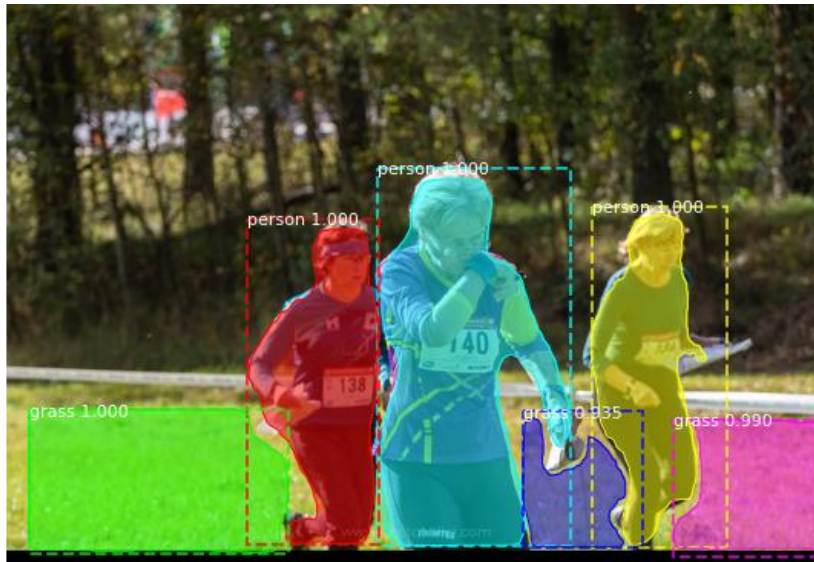
Attēlā 4.16. darba autora uztrenētais modelis atrada visus cilvēkus, visas sejas un arī sienas banneri, kas palīdz noteikt atrašanās vietu (apbalvošanas zona). Cilvēku un seju maskas tika iezīmētas diezgan precīzi. Bannera siena tika segmentēta ar diviem objektiem. Kaut arī patiesībā ir tikai viens sienas banneris, augsta līmeņa klasificēšanai ir nepieciešams vismaz viens šāds objekts, tāpēc šajā gadījumā tā nav problēma. Bannera sienas maskas patiesībā arī ir iezīmētas diezgan korekti. Var pamanīt, ka tās nedaudz pārklājās.



4.17. att. Dalībnieki un skatītāji blakus stāvvietas zonai

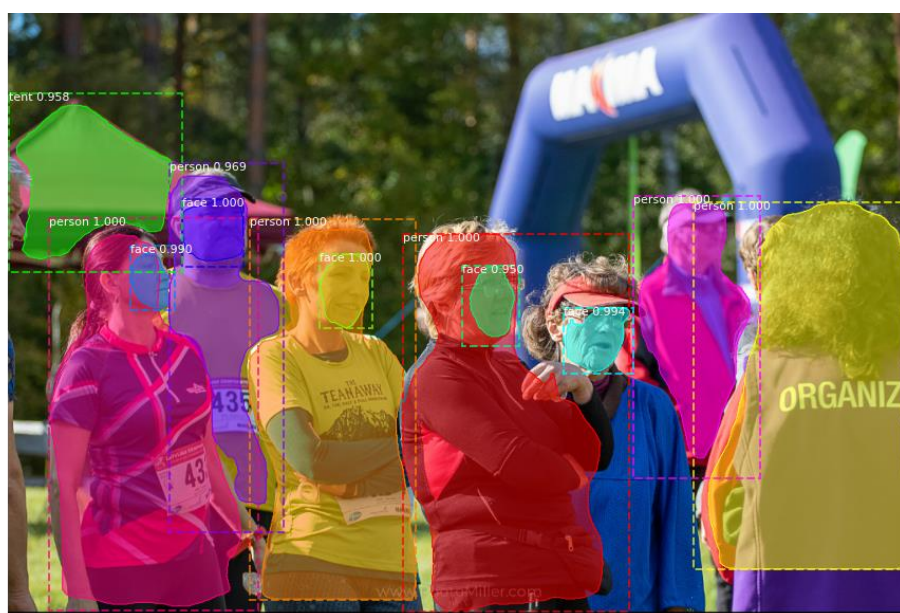
Attēlā 4.17. darba autora uztrenētais modelis atrada gandrīz visus labi saskatāmus cilvēkus, taču tikai 1 seju. Stāvvietā ir diezgan daudz automašīnu, taču tika atrastas tikai 4, bet dotajā fotogrāfijā ar to pietiek, lai secinātu, ka ar lielu iespējamību cilvēki atrodas stāvvietas

zonā vai blakus tai. Šajā gadījumā ar mašīnām ir tāda pati situācija kā sadaļa 4.6.1. ar cilvēkiem. Labi saredzami auto ir atrasti un iezīmēti, taču automašīnas, kuru tikai jumti ir redzami, netika atrasti. Tas ir tāpēc, ka izvēlētajā fotogrāfiju kolekcijā, kura satur 1172 fotogrāfijas, atrodas tikai aptuveni 15 fotogrāfijas, kurās ir labi redzama stāvvietā ar mašīnām. Visas segmentācijas maskas ir ļoti labi iezīmētas.



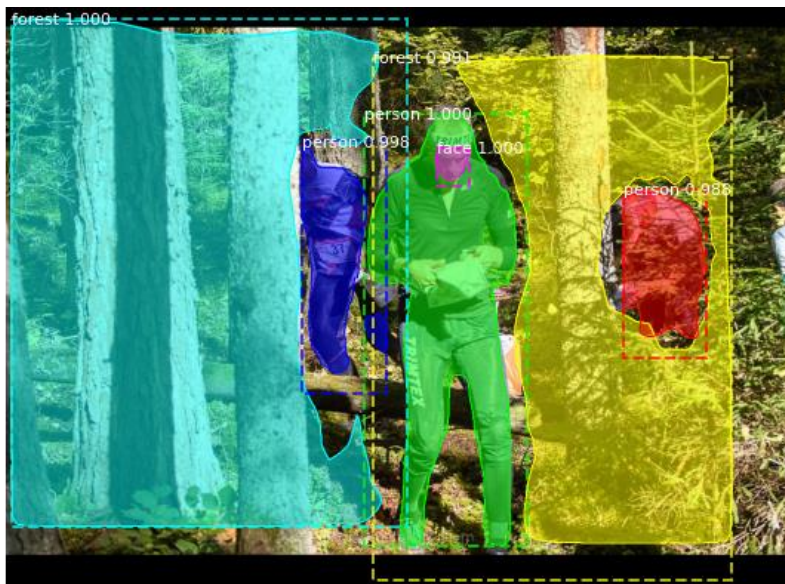
4.18. att. Sacensību dalībnieki zālājā

Attēlā 4.18. darba autora uztrenētais modelis atrada vairākus zāles gabalus un ļoti labi iezīmēja segmentācijas maskas. Tika atrasti un segmentēti arī visi 3 cilvēki, kuri ir redzami no priekšas, taču nevienu seju modelis neatrada, kaut arī sejas ir diezgan labi saskatāmas (tās nav pārāk mazas).



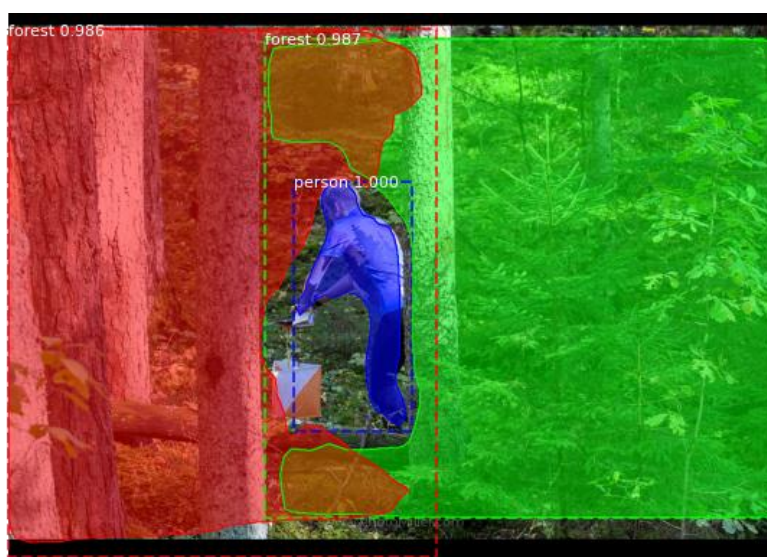
4.19. att. Sacensību dalībnieki un organizators

Pretēji attēlā 4.18. redzamajai nesekmīgai seju noteikšanai, attēlā 4.19. var pamanīt, ka uztrenētais modelis atrada un segmentēja gandrīz visas sejas un gandrīz visus cilvēkus; grūti pateikt, kāpēc attēlā 4.18. modelis neatrada nevienu seju. Tika atrasts arī atrašanās vietu apzīmējošs objekts – āras tents. Segmentāciju maskas ir iezīmētas labi.



4.20. att. Sacensību dalībnieki mežā

Attēlā 4.20. tika korekti atrasti un iezīmēti visi objekti: 3 cilvēki, 1 redzamā seja un mežs. Šajā gadījumā izceļas tas, ka tika atrasts cilvēks, kas atrodas aiz egles – šo cilvēku ir daudz grūtāk saskatīt, taču modelis tāpat spēja noteikt cilvēku aprakstošās iezīmes. Tika atrastas 2 meža instances; viena segmentācijas maska ir iezīmēta labi, otra – gandrīz labi, jo tika izlaists neliels meža gabals pa labi, taču jāievēro, ka instanču segmentēšanas neiekļāva vietu, kurā ir redzams cilvēks aiz egles, un ka tas gandrīz vai perfekti atkārtu 2. cilvēka (cilvēks pa vidu) kontūru.



4.21. att. Sacensību dalībnieks mežā

Attēlā 4.21. modelis atrada cilvēku un mežu, kas ir vienīgi objekti, kuri apzīmē attēlā notiekošo. Semantiskās maskas ir iezīmētas ļoti labi.

Kopumā darba autora uztrenētais instanču segmentēšanas modelis spēj atrast lielāko daļu fotogrāfijā attēloto objektu, protams, gadās arī tā, ka pat labi saskatāmus objektus modelis nespēj noteikt un segmentēt; iespējams, tādos gadījumos modelis nespēja atrast objektu specifiskās iezīmes mazas treniņkopas dēļ (tas var būt rakursu vai apgaismojuma dēļ). Var pamanīt, ka lielākā daļa segmentācijas masku ir iezīmētas ļoti labi, tas var palīdzēt kādos konkrētos gadījumos salīdzināt segmentācijas masku izmērus, lai noteikt, piemēram, kura atrašanās vieta dominē attēlā jeb kura atrašanās vieta aizņem lielāku fotogrāfijas daļu jeb laukumu. Izvēlēto fotogrāfijas kolekciju, uz kuras tika trenēts modelis, tas segmentēja un noteica objektu instances labi, ņemot vērā to, ka tika izmantotas tikai 96 fotogrāfijas 10 klašu trenēšanai.

4.6.3. Rezultāti no citas maratona fotogrāfiju kolekcijas

Tika izvēlēta vēl viena fotogrāfiju kolekcija, kura arī attēlo mežā notiekošu maratonu [42], taču tas notika citā dienā, citā vietā (arī mežā, bet ne tajā pašā, kas 1.izvēlētajā fotogrāfiju kolekcijā) un ir attēloti citi cilvēki, kā arī vairākās fotogrāfijās ir pamanāmi citi fotografēšanas rakursi. Testēšanas nolūkos izvēlētā cita fotogrāfiju kolekcija satur 210 fotogrāfijas un tās autors ir Mareks Gaļinovskis.

Uztrenētā instanču segmentēšanas modeļa testēšana uz citas fotogrāfiju kolekcijas var palīdzēt noteikt, vai uztrenētais modelis var tikt izmantots kā universāls instanču segmentēšanas modelis līdzīga veida maratoniem, kur dalībnieki skrien pa meža takām nevis pa pilsētas ceļiem.



4.22. att. Sacensību dalībnieki pasākuma notikuma vietā

Attēlā 4.22. tika atrasta un labi iezīmēta zāle, atrasti gandrīz visi labi redzami cilvēki un arī vairāku cilvēku sejas. Kā arī tika atrasta viena no trijām tentīm un viens no trijiem pludmales karogiem. Pietiek ar vienu atrašanās vietu apzīmējošu objektu, lai noteiktu konkrēto atrašanās vietu; skaits var palīdzēt noteikt, piemēram, cik publiska vieta tā ir.



4.23. att. 3 dalībnieki fotografēti no priekšas

Attēlā 4.23. tika atrasts viens pludmales karogs, kas apzīmē galveno pasākuma notikuma vietu, kā arī tika atrasta zāle. Visi 3 cilvēki un viņu sejas arī tika korekti noteiktas un labi iezīmētas.



4.24. att. Viens dalībnieks mežā

Attēlā 4.24. tika atrasts viens cilvēks un viņa seja. Kā arī tika atrasts un iezīmēts gandrīz viss mežs; tā segmentēšanas maska nav perfekta, taču lielāko daļu tas iezīmēja korekti.



4.25. att. Sacensību dalībnieki apbalvošanas zonā

Attēlā 4.25. tika atrasti visi cilvēki, 2 sejas un tikai 1 atrašanās vietu apzīmējošs objekts – pjedestāls, kaut arī ir redzami arī tādi objekti kā bannera siena, 2 pludmanles karogi un tents.



4.26. att. Sacensību dalībnieki apbalvošanas zonā

Attēlā 4.26., tāpat kā attēlā 4.25., ir attēloti dalībnieki apbalvošanas zonā, taču attēlā 4.26. tika atrasti vairāki atrašanās vietu apzīmējoši objekti (pludmales karogi, sienas banneris un zāle); tiešās atrašanās vietas (apbalvošanas zona) apzīmējošie objekti šajā gadījumā ir bannera siena un pjedestāls – kaut arī netika atrasts pjedestāls, tika atrasta bannera siena, kura tāpat apzīmē apbalvošanas zonu, tāpēc šajā gadījumā ir iespējams korekti noteikt atrašanās vietu (jo vairāk ir konkrētās atrašanās vietu apzīmējošie objekti, jo lielāka iespējamība, ka tiks korekti noteikta fotogrāfijā attēlotā specifiskā atrašanās vieta). Netika atrasts arī 1 cilvēks un netika atrasta neviena seja. Bannera sienas segmentācijas maska ir iezīmēta slikti, jo var redzēt, ka tā

tika iezīmēta pa virsu cilvēkiem, taču galvenais klasificēšanas uzdevumā ir noteikt objektus, kas šajā gadījumā tika izdarīts labi.

4.7. Fotogrāfijā attēlotā notikuma klasificēšanas algoritms

Viena fotogrāfija var attēlot vairākus objektus, kuri apraksta dažādas atrašanās vietas, tāpēc ir jāveic atrašanās vietu prioritizēšana jeb jānosaka, kuras atrašanās vietas apraksta specifisku notikumu. Ja fotogrāfijām piešķirt visas atrastās atrašanās vietas, tad šis vairāk būtu fotogrāfiju tagošanas modelis nevis augsta līmeņa notikumu klasifikators.

Klasifikators izdod 3 atbildes: atrašanās vietu, cilvēku grupas lielumu (viens cilvēks, divi cilvēki, maza grupa, liela grupa, pūlis) un cilvēku novietojumu pret kameru (ar sejām vai muguru). Visas 3 atbildes kopā apraksta specifisku fotogrāfijā attēloto notikumu.

4.7.1. Klasificēšanas algoritma darbības apraksts

Lai būtu iespējams noteikt fotogrāfijā attēloto notikumu jeb veikt augsta līmeņa klasificēšanu, nepietiek tikai noteikt fotogrāfijās attēlotos objektus. Ir nepieciešams saprast objektu savstarpējo novietojuma un esamības saistību. Dažādi objekti pa vienam var apzīmēt dažādas notikuma vietas; katrā notikuma vietā var tik veiktas specifiski notikuma vietai piederošas aktivitātes. Vairāku objektu kopums jeb apvienojums var sekmēt vienlīdz nozīmīgu notikuma vietas apzīmēšanu (piemēram, tabulā 4.1. definētās kategorijas “sienas banneris”, “pludmales karogs” un “nožogojuma banneris” visas apzīmē vienu un to pašu notikuma vietu – “Galvenā pasākuma notikuma vieta”), bet ir iespējams, ka objektu kopā ir tādi objekti, kuri ar augstāku prioritāti nosaka atrašanās vietu. Piemēram, pjedestāls nosaka to, ka atrašanās vieta ir apbalvošanas zona neatkarīgi no tā, vai tika atrasti, piemēram, pludmales karogi vai automašīnas, jo pjedestāls parasti nekad neatrodas stāvvietas zonā vai galvenajā pasākuma notikuma vietā, kur nav paredzēta apbalvošana.

Lai sasaistītu atrastos objektus ar tabulā 4.1. definētajām kategorijām un izveidotu konkrētu fotogrāfiju aprakstošu augsta līmeņa kategoriju, tika uzrakstīta funkcija “classifyImage()” (sk. 1.pielikumu). Tā saņem 3 argumentus: Mask R-CNN konkrētam attēlam iegūtos rezultātus (atrstos objektus, maskas, varbūtības, u.c.) un lielas cilvēku grupas definēšanu jeb robežas (minimālais cilvēku skaits lielā grupā un maksimālais cilvēku skaits lielā grupā). Lielas grupas definēšana ļauj izveidot daudz specifiskāku klasifikatoru – var klasificēt fotogrāfijas balstoties uz attēloto cilvēku daudzumu fotogrāfijā; piemēram, ja tiktu definētas tikai tādas klases kā “viens cilvēks”, “divi cilvēki” un “cilvēku grupa”, tad gadījumos, kad fotogrāfijā cilvēku skaits ir lielāks par 2, visas fotogrāfijas informētu tikai par to, ka attēlā

ir redzama cilvēku grupa, bet tā ir ļoti vispārīgā informācija, jo, iespējams, lietotājs grib atrast tikai tādas fotogrāfijas, kurās ir attēlotas cilvēku grupas līdz 5 cilvēkiem nevis milzīgu cilvēku pūli. Šī pieeja nodrošina iespēju veikt specifiskāku klasificēšanu, tātad dodot vairāk konteksta par fotogrāfiju. Lielas grupas definēšanas robežas ir nepieciešamas tādēļ, ka katrs cilvēks jēdzienus “liela grupa” vai “maza grupa” uztver subjektīvi – kāds uzskata, ka maza grupa satur maksimāli 5 cilvēkus, kāds cits var uzskatīt, ka maza grupa pasākumos maksimāli satur 10 cilvēkus, – nav konkrēta standarta, kurš nosaka, ko nozīmē liela vai maza grupa; tas ir subjektīvs interpretējums. Tādējādi, subjektīvi definējot tikai lielas grupas robežas, tiek definētas arī mazas grupas robežas un cilvēku pūļa robežas – ja, piemēram, lielas grupas robežas ir [7, 15], tad mazas grupas robežas būs [3, 7) un cilvēku pūļa robežas būs (15, +∞).

Tā kā Mask R-CNN atgriež nevis klašu nosaukumus, bet klašu identifikatorus (id), tad sākumā tika inicializēta vārdnīca “classDict”, kur savieno klašu jeb kategoriju nosaukumus ar tiem piederošiem identifikatoriem.

Tālāk tika definētas atrašanās vietas kategoriju apzīmējošo objektu prioritātes, kuras palīdzēs noteikt vispiemērotāko jeb specifiskāko attēloto atrašanās vietu fotogrāfijā. Tika noteikts, ka ar visaugstāko prioritāti ir atrašanās vieta “apbalvošanas zona”, jo tā paskaidro konkrēti ar sporta sacensībām saistītu notikumu – apbalvošanu. Tālāk seko atrašanās vieta “stāvvietā”, jo vienīgā vieta dotajās sacensībās, kur ir redzams liels automašīnu skaits, ir stāvvietā. Pēc atrašanās vietas “stāvvietā” seko “Galvenā pasākuma notikuma vieta”, kuru apzīmē tādi objekti kā pludmales karogi, nožogojuma banneri un āras tentis jeb nojumes. Pēc atrašanās vietas “Galvenā pasākuma notikuma vieta” seko “Zālājs”, jo atrastā meža instance, iespējams, attēlo mežu kā fonu, un atrašanās vieta ar viszemāko prioritāti ir “Mežs”, jo tajā neatrodas nevienas citas atrašanās vietu apzīmējoši objekti (vienīgi cilvēki un sejas, taču šie objekti nenosaka atrašanās vietas kategoriju). Klasēm “cilvēks” un “seja” arī tika pievienotas prioritātes, taču tās ir domātas nevis atrašanās vietas noteikšanai, bet atrasto klašu saraksta elementu ierēšanas secības definēšanai.

Tādām kategorijām kā “cilvēks”, “seja” un “automašīna” ir nepieciešams zināt attēloto objektu skaitu, jo no tā ir atkarīgi tādi kategoriju viedi kā “cilvēku grupu lielums” (t.sk. kategorijas “viens cilvēks” un “divi cilvēki”), “cilvēku novietojums pret fotokameru”, kā arī no objektu skaita ir atkarīga atrašanās vietas kategorija “stāvvietas zona”. Cilvēku grupu lieluma kategorijas tiek definētas, balstoties uz funkcijai “classifyImage” padoto lielās grupas definēšanas intervālu. Cilvēku novietojums pret kameru balstās uz atrasto cilvēku un seju skaitu fotogrāfijā. Atrašanās vieta “stāvvietā” balstās uz atrasto automašīnu skaitu (vairākās fotogrāfijās bija redzamas 1 vai 2 automašīnas, taču tās, iespējams, bija organizatoru vai piegādes transports, kas atradās pasākuma galvenajā notikuma vietā nevis stāvvietā, tāpēc tika

pieņemts, ka, ja ir atrastas vismaz 3 automašīnas, tad ar lielu varbūtību fotogrāfija varētu attēlot, piemēram, cilvēkus vai kādus citus objektus stāvvietā vai blakus tai).

Lai iegūtu atrasto objektu skaitu katrai klasei, no Mask R-CNN rezultātu saraksta tika izveidots jauns saraksts ar pāriem, kur katrs pāris satur atrastās klases identifikatoru (vienu reizi; katrai klasei savs pāris) un dotās klases atrasto objektu skaitu. Pēc tam tiek izmantots cikls, kurš katram pārim pievieno vēl vienu elementu, kas ir prioritāte. Kad ir izveidots jauns saraksts, kurš satur kortežus (angliski: tuples) ar 3 elementiem, saraksts tiek kārtots pēc kortežu pēdējā elementa (prioritātes) dilstošā secībā, tādējādi objektu klases ar zemāko prioritāti tiek apstrādātas pirmās.

Sākumā tiek veikta cilvēku skaita noteikšana, tāpēc klases “cilvēks” jeb “person” prioritāte ir viszemākā. Balstoties uz definētajām lielas grupas robežām un atrasto cilvēku skaitu fotogrāfijā, tiek piešķirta kāda no sekojošām kategorijām: “Viens cilvēks”, “Divi cilvēki”, “Maza cilvēku grupa”, “Liela cilvēku grupa” un “Cilvēku pūlis”. Nākamā pēc prioritātes tiek apskatīta klase “seja”, jo cilvēku novietojumu pret kameru var noteikt, zinot atrasto cilvēku un atrasto seju skaitu fotogrāfijā. Tā kā treniņkopa ir patiešām maza, modelis nespēja perfekti iemācīties atpazīt sejas – tas spēja labi atpazīt sejas tuvumā, taču sejas tālumā atpazīna ar grūtībām (tas ir arī tāpēc, ka sejas tālumā ir ļoti mazas un ir grūti noteikt sejām piemītošas iezīmes). Ņemot vērā šo uztrenētā modeļa trūkumu, lai noteiktu to, ka cilvēki fotogrāfijā ir vairāk vērsta pret kameru ar sejām nevis ar muguru, pieaugot cilvēku skaitam, tika samazināts nepieciešamais seju skaits. Pēc loģikas var spriest, ka, ja vismaz puse no cilvēkiem ir vērsta pret kameru ar sejām, tad cilvēku grupa nav fotografēta no muguras. Jo vairāk cilvēku tiek ietilpināti fotogrāfijā, jo tālāk tie atrodas no fotogrāfa (cilvēkus paliek grūtāk saskatīt, t.sk. arī cilvēku sejas). Tāpēc eksperimentāli tika izsecināts, ka, lai klasifikators spētu korekti klasificēt izvēlēto fotogrāfiju kolekciju (ņemot vērā to, ka darba autora uztrenētajam modelim ir grūtības noteikt un segmentēt sejas tālumā), kategorijas veida “cilvēku novietojums pret kameru” noteikšanai tika definēti sekojoši nosacījumi:

- Ja cilvēku skaits fotogrāfijā ir mazāks par 11, tad cilvēku grupa ir vērsta ar sejām pret kameru, ja atrasto seju skaits ir $\text{cilvekuSkaits}/2$ vai lielāks;
- Ja cilvēku skaits fotogrāfijā ir lielāks vai vienāds ar 11, tad cilvēku grupa ir vērsta ar sejām pret kameru, ja atrasto seju skaits ir $\text{cilvekuSkaits}/3$ vai lielāks;
- Ja cilvēku skaits fotogrāfijā ir lielāks vai vienāds ar 20, tad cilvēku grupa ir vērsta ar sejām pret kameru, ja atrasto seju skaits ir $\text{cilvekuSkaits}/4$ vai lielāks;
- Ja cilvēku skaits fotogrāfijā ir lielāks vai vienāds ar 25, tad cilvēku grupas novietojums pret kameru netiek uzstādīts, jo cilvēki ir ļoti mazi un to sejas gandrīz vai nevar saskatīt, tāpēc nevar ar lielu varbūtību pateikt, vai cilvēku pūlis ir fotografēts no priekšas vai

muguras.

Protams, šie nosacījumi ir diezgan subjektīvi, jo katrs cilvēks var dažādi uztvert apgalvojumu “grupa fotografēta no priekšas”. Kāds var uzskatīt, ka šis apgalvojums ir patiess tikai tad, ja pilnībā visi cilvēki ir fotografēti no priekšas; kāds cits var uzskatīt, ka dotais apgalvojums ir patiess, ja 75% cilvēku ir vērsti ar sejām pret kameru.

Kad ir noteikti kategoriju veidi “cilvēku grupas lielums” un “cilvēku novietojums pret kameru”, nākamais solis ir atrašanās vietas noteikšana. Prioritātes sistēma šajā gadījumā palīdz noteikt atrašanās vietu, kura sniedz visvairāk informācijas jeb kontekstu. Tā kā objekti, kuri apzīmē konkrētas atrašanās vietas, tika prioritizēti un sakārtoti pēc prioritātes dilstošā secībā (zemākā prioritāte pirmā), tad atrašanās vietas noteikšana notiek sekojošā veidā – pieņemsim, ka tika atrasti tādi objekti jeb instances kā tents, mežs, pjeDESTāls un 3 automašīnas. Veicot prioritizēšanu, tiek izveidots sakārtots saraksts [(“mežs”, 1), (“tents”,1), (“automašīna”, 3), (“pjeDESTāls”, 1)]. Sākumā mainīgajam, kurš atbild par atrašanās vietas kategorijas vērtību (“location”), tiek piešķirta vērtība “Nezināma atrašanās vieta”, gadījumā, ja neviens atrašanās vietu apzīmējošs objekts netika atrasts. Tad iet cauri sakārtotā saraksta pārišiem: pirmais ir “mežs”, tāpēc mainīgā “location” vērtība tiek nomainīta uz “Mežs”; klasificēšanas algoritmā tika izmantoti angļu kategoriju nosaukumi, jo tādā gadījumā nerodas problēmas ar locījumiem, apvienojot visus 3 kategoriju veidus vienā – veidojas saprotama un gramatiski korekta kategorija. Nākamais objekts ir “tents”, kas apzīmē galveno pasākuma notikuma vietu, tāpēc mainīgajam “location” tiek piešķirta vērtība “Galvenā pasākuma notikuma vieta”. Līdz pat saraksta beigām tiek veikta šī piešķiršanas/aizvietošanas darbība, lai beigās mainīgais “location” saturētu to atrašanās vietu, kura visprecīzāk apzīmē fotogrāfijā redzamo atrašanās vietu.

Pēc visu atrasto objektu pēcapstrādes, tika veikta izņēmumgadījumu apstrāde; ir iespējams, ka fotogrāfijā nav neviena cilvēka vai nevienas sejas (ja cilvēki ir fotografēti no muguras). Tāpēc, ja atrastais seju skaits ir 0 un attēlā ir no 1 līdz 24 cilvēkiem (ieskaitot), tad tiek pieņemts, ka attēlā redzami cilvēki ir fotografēti no muguras jeb vērsti ar muguru pret fotogrāfu. Ja fotogrāfijā vispār nav cilvēku, tad cilvēku novietojums pret kameru netiek inicializēts jeb algoritmā – tukša simbolu virkne. Var arī gadīties, ka instanču segmentēšanas modelis atrada seju, bet neatrada cilvēku, tāpēc ir iespējams, ka seju skaits ir lielāks par atrasto cilvēku skaitu. Šādā gadījumā cilvēku grupas lieluma kategoriju nosaka atrastais seju skaits.

4.8. Klasifikatora testu rezultāti

Lai pārbaudītu, cik labi uztrenētais instanču segmentēšanas modelis, darbojoties kopā ar klasificēšanas algoritmu, spēj veikt augsta līmeņa klasificēšanu gan uz fotogrāfiju kolekcijas, kura tika izmantota modeļa trenēšanai, gan uz citas, līdzīgas maratonu attēlojošas kolekcijas, no katras kolekcijas tika izvēlēta 41 fotogrāfija, kur fotogrāfijas attēlo dažādas atrašanās vietas un dažādu cilvēku un citu objektu skaitu. Tika izveidota vārdnīca, kura savienoja fotogrāfiju identifikatorus ar patieso, manuāli pierakstīto attēlā redzamo notikumu. Palaižot kodu, tika atgriezts fotogrāfijā attēlotais patiesais notikums (no vārdnīcas) un klasificēšanas algoritma prognozētais notikums. Tad abi notikumi tika salīdzināti, lai noteiktu, kurus no trijiem kategoriju veidiem klasificēšanas algoritms ir prognozējis korekti.

4.8.1. Izvēlētās fotogrāfiju kolekcijas klasificēšanas testēšanas rezultāti

4.5. tabula.

Klasifikatora testu rezultāti uz izvēlētās fotogrāfiju kolekcijas

Novērtēšanas metrika	Korekti prognozētas fotogrāfijas	Precizitāte
Tikai cilvēku grupu lielums	38/41	93%
Tikai atrašanās vieta	35/41	85%
Tikai cilvēku novietojums pret kameru	37/41	90%
Visi 3 kategoriju veidi kopā	28/41	68%

Kaut arī tika izveidota maza testēšanas kopa (41 fotogrāfija), no iegūtajiem rezultātiem, kuri ir attēloti tabulā 4.5., var secināt, ka augsta līmeņa klasifikators spēj ar diezgan augstu precizitāti korekti prognozēt katru no trijiem kategorijas veidiem atsevišķi. Var pamanīt, ka vissliktāk klasifikators spēj noteikt atrašanās vietas kategoriju. Tas ir tāpēc, ka, pirmkārt, atrašanās vietas kategorijas nosaka 8 dažādas klases jeb 8 objektu tipi (ja ir atrasti vairāki objekti, kur katrs apzīmē definētās atrašanās vietas, taču konkrētās fotogrāfijā attēlotās atrašanās vietas apzīmējošs objekts netika atrasts, tad attēlotā atrašanās vietas kategorija netiek korekti noteikta), otrkārt, treniņkopā gandrīz visās fotogrāfijās bija attēloti cilvēki un cilvēku sejas no dažādiem rakursiem (tāpēc klases “cilvēks” un “seja” segmentēšanas modelis iemācījās labāk). Mežs un zāle arī bija lielākajā daļā fotogrāfiju, taču pārējās 6 klases, kuras arī tiek izmantotas atrašanās vietas kategorijas noteikšanā, bija redzamas tikai nelielā treniņkopas daļā jeb katrai no 6 klasēm bija no 10 - 30 trenēšanai pieejamo paraugu – tātad mazāk nekā klasēm “cilvēks” un “seja”. Var arī pamanīt, ka visus 3 kategoriju veidus kopā tas prognozē ar daudz zemāku precizitāti – 68%. Tas ir tāpēc, ka vienā fotogrāfijā tas, iespējams, korekti prognozēja

cilvēku grupas lielumu, bet nekorekti – atrašanās vietu, un citā fotogrāfijā, piemēram, otrādi. Tātad šajos abos gadījumos klasifikators neuzminēja pilnībā visu fotogrāfijā attēloto notikumu (korekti prognozēja tikai daļu no visa patiesā notikuma).

Kopumā visu 3 kategorijas veidu korekta prognozēšana (korekti prognozēts fotogrāfijā attēlotais notikums) ar precizitāti 68% ir diezgan labs rezultāts, ņemot vērā to, ka instanču segmentēšanas modelis, kura atgrieztos rezultātus izmanto klasificēšanas algoritms, tika trenēts uz treniņkopas ar tikai 96 fotogrāfijām un tas tika trenēts uz 10 dažādām klasēm.

4.8.2. Citas, pēc satura līdzīgas fotogrāfiju kolekcijas testēšanas rezultāti

Tāpat kā instanču segmentēšanas modeļa testēšanā, klasifikatora testēšanā arī tika izmantota tā pati cita, līdzīga fotogrāfiju kolekcija (kura arī attēlo maratonu), lai noteiktu, cik precīzi augsta līmeņa klasifikators spēs noteikt korekti fotogrāfijās attēloto notikumu, ņemot vērā to, ka instanču segmentēšanas modelis netika trenēts uz šīs (citas) kolekcijas fotogrāfijām.

4.6. tabula.

Klasifikatora testu rezultāti uz citas, līdzīgas fotogrāfiju kolekcijas

Novērtēšanas metrika	Korekti prognozētas fotogrāfijas	Precizitāte
Tikai cilvēku grupu lielums	34/41	83%
Tikai atrašanās vieta	28/41	68%
Tikai cilvēku novietojums pret kameru	31/41	76%
Visi 3 kategoriju veidi kopā	18/41	44%

Tabulā 4.6. var pamanīt, ka katra kategoriju veida precizitāte atsevišķi ir kritusies par aptuveni 13.6%. Vissliktāk, tāpat kā izvēlētās fotogrāfiju kolekcijas rezultātos, tika noteikta atrašanās vietas kategorija. Kategorijas veidi “Cilvēku grupas lielums” un “Cilvēku novietojums pret kameru” varētu būt samazinājies šīs citas fotogrāfijas kolekcijas rakursu dēļ. Vairākās fotogrāfijās tika izmantots platleņķa objektīvs, kurš padara cilvēkus, tai skaitā viņu sejas, daudz mazākas, tāpēc tās ir grūtāk saskatīt. Kā arī šajā fotogrāfiju kolekcijā ir daudz vairāk fotogrāfiju, kuras satur lielu cilvēku skaitu. Tas apgrūtināja darba autora manuālu konkrēto patieso notikumu, jo ir grūti saprast, kuros gadījumos, piemēram, iekļaut cilvēkus fonā, un kuros nē.

Protams, kopumā arī rezultāts ir sliktāks uz šīs fotogrāfiju kolekcijas, jo tajā bija izmantoti citi fotografēšanas rakursi, ir attēloti vairāki citi objekti (tai skaitā mežs, kurā koki nav tik cieši blakus viens otram, kā tas ir izvēlētajā fotogrāfiju kolekcijā, uz kuras instanču segmentēšanas

modelis tika trenēts); iespējams arī gaismas krišanas leņķis varētu būt viens no faktoriem, kas samazināja klasifikatora precizitāti uz šīs fotogrāfiju kolekcijas (fotogrāfiju kolekcija, uz kuras tika trenēts instanču segmentēšanas modelis, satur specifiskas, iespējams, sīkas atšķirības, kuras tāpat ietekmē darba autora uztrenētā modeļa spēju segmentēt fotogrāfijas no citām kolekcijām, kuras netika izmantotas trenēšanā).

SECINĀJUMI

Sporta sacensību fotogrāfijas var klasificēt vairākos veidos. Var klasificēt fotogrāfijas tikai pēc attēlotajiem sporta veidiem. Var klasificēt viena konkrēta sporta veida fotogrāfijas, balstoties uz dotā sporta veida aktivitātēm un atrašanās vietām. Var arī klasificēt fotogrāfijas, ņemot vērā fotografēšanas rakursus (tuvplāna kadrs, kadrs no maza attāluma, kadrs no liela attāluma).

Maratonu attēlojošā fotogrāfiju kolekcija, kura tika izvēlēta, lai izstrādātu dotās un tai līdzīgu kolekciju augsta līmeņa klasifikatoru, tika aprakstīta un izanalizēta. Darba autors secina, ka maratonos netiek izmantots kāds specifisks inventārs, kas apzīmētu tieši ar maratonu saistītu sporta notikumu. Izvēlētās fotogrāfiju kolekcijas analizēšanas rezultātā tika noskaidrots, ka doto kolekciju var klasificēt, balstoties uz 3 dažādiem kategoriju veidiem: cilvēku grupu lielums, cilvēku vai cilvēku grupas novietojums pret kameru un atrašanās vieta. Kombinējot visus trīs kategoriju veidus, tiek iegūta specifiska, konkrētu notikumu aprakstoša klasificēšanas kategorija.

Attēlu segmentēšana sniedz ļoti detalizētu informāciju par fotogrāfijām, jo katra fotogrāfija tiek sadalīta vairākos segmentos, kurus var izmantot, lai saprastu attēloto apkārtējo vidi un attēloto objektu savstarpējo saistību, tādējādi ir iespējams noteikt detalizētu informāciju par attēloto notikumu. Attēlu segmentēšana var palīdzēt definēt un klasificēt patiešām sarežģītus un specifiskus notikumus fotogrāfiju – var izmantot segmentāciju masku laukumus, robežas un savstarpējo novietojumu, lai, piemēram, noteikt, kurš objekts aizņem lielāku attēla daļu, tādējādi iegūstot fotogrāfijā dominējošo objektu.

Kaut arī instanču segmentēšanas modeļi ir domāti, lai darbotos ar saskaitāmiem objektiem jeb atsevišķām instancēm, instanču segmentēšanas modeļa trenēšanas procesā tika noskaidrots, ka Mask R-CNN ietvars, kas ir instanču segmentēšanas ietvars, spēj fotogrāfijā noteikt un iezīmēt ne tikai saskaitāmus objektus, bet arī tādus nesaskaitāmos objektus kā mežu un zāli, tādējādi nosakot fotogrāfijās attēloto fonu. Šajā gadījumā ir iespējams, ka modelis var atrast vairākas meža un zāles instances (kaut arī tie ir nesaskaitāmi objekti), taču definētajā klasificēšanas uzdevumā galvenais ir noteikt konkrētās atrašanās vietu apzīmējošu objektu esamību nevis skaitu.

Uz COCO datukopas iepriekš apmācīts instanču segmentēšanas modelis ir ļoti labi pielāgojams citām datu kopām. Izvēlētās fotogrāfiju kolekcijas analizēšanas fāzē tika noteikts, ka dotajā maratona kolekcijā ir 10 dažādi objekti jeb klases, kuras var noteikt kādu konkrētu fotogrāfiju aprakstošu kategoriju. Instanču segmentēšanas modelis tika uztrenēts uz visām 10 klasēm ar ļoti mazu treniņkopas izmēru (96 fotogrāfijas), taču uztrenētais modelis veica

instanču segmentēšanu diezgan labi. Kaut arī iepriekšapmācīts modelis bija apmācīts uz cilvēku un automašīnu atpazīšanu, kura arī bija nepieciešama dotās maratona fotogrāfiju kolekcijas klasificēšanai, veicot ar TL tehnikas palīdzību modeļa trenēšanu uz jaunām definētajām klasēm, t.sk. uz cilvēkiem un automašīnām, instanču segmentēšanas modelis ir atkal jātrenē uz cilvēku un automašīnu atpazīšanu, jo iepriekšapmācīta modeļa klasificēšanas slānis tiek aizvietots ar jaunu klasificēšanas slāni – tādējādi tiek pazaudēta saikne starp jaunā klasificēšanas slāņa klasēm “cilvēks” un “automašīna” un modeļa apgūtajām zināšanām par cilvēkiem un automašīnām.

Tika noskaidrots, ka darba autora uztrenētā instanču segmentēšanas modeļa lielākie trūkumi ir cilvēku seju noteikšana gadījumos, kad cilvēki atrodas diezgan tālu no fotogrāfa (sejas ir mazas), cilvēku instanču noteikšana gadījumos, kad ir redzamas tikai cilvēku galvas vai cilvēki līdz pleciem, un automašīnu noteikšana, ja ir redzami tikai automašīnu jumti. Visi šie trūkumi varētu būt saistīti ar mazo treniņkopas izmēru. Automašīnu trenēšanas gadījumā tas ir ļoti iespējami, jo izvēlētajā fotogrāfiju kolekcijā, kurā ir 1172 fotogrāfijas, tikai 15 fotogrāfijas attēlo kaut cik labi saredzamas automašīnas, taču no 15 fotogrāfijām tikai 8 tika izmantotas trenēšanai, lai tiktu atstātas fotogrāfijas arī testēšanai.

Uzrakstītais klasificēšanas algoritms, kas balstās uz instanču segmentēšanas modeļa atgrieztajiem rezultātiem par atrastajiem objektiem, tika testēts gan uz izvēlētajās fotogrāfiju kolekcijas, gan uz citas, līdzīgas fotogrāfiju kolekcijas. Trenēšanā izmantotās kolekcijas fotogrāfijas klasifikators spēja pilnībā noteikt korekto attēloto notikumu 68% gadījumu. Tas ir labs rezultāts, ņemot vērā to, ka klasifikatoram ir korekti jāprognozē visi 3 kategoriju veidi. Trenēšanā neizmantotās kolekcijas fotogrāfijas klasifikators spēja pilnībā noteikt korekto attēloto notikumu tikai 44% gadījumu. No šiem datiem var secināt, ka darba ietvaros izveidotais klasifikators, kurš tika trenēts uz vienas, konkrētas fotogrāfiju kolekcijas, nav pielietojams, lai korekti klasificētu citu, līdzīgu fotogrāfiju kolekciju. Tas varētu būt vairāku faktoru dēļ: maza treniņkopa augsta līmeņa klasificēšanas uzdevumam, citā fotogrāfiju kolekcijā ir dažādi citi objekti, kuri mulsina uztrenēto instanču segmentēšanas modeli, katru fotogrāfiju kolekciju fotografēja cits fotogrāfs – katram fotogrāfam ir savs fotografēšanas stils un tehnika, tāpēc abās kolekcijās ir novērojami dažādi fotografēšanas rakursi un objektīvu pielietojumi – rakursi var nedaudz izmantīt attēloto objektu formu vai attēlot vienus un tos pašus objektus no dažādiem skatupunktiem (no katra skatupunkta objekts izskatās nedaudz citādi, tātad attēlotās iezīmes arī mainās). Tā kā meži var būt dažādu tipu un veidu, tas arī ietekmē konkrētā notikuma atpazīšanu.

Mask R-CNN ir labs instanču segmentēšanas ietvars, ar kura palīdzību var atrast un segmentēt fotogrāfijā attēlotos objektus, tādējādi iegūstot detalizētu informāciju par fotogrāfijā attēloto notikumu, kuru var izmantot paša veidots klasificēšanas algoritms konkrētu notikumu

klasificēšanai. Taču, lai izveidotu klasifikatoru, kuru varētu izmantot, piemēram, tieši mežā notiekošu maratona fotogrāfiju kolekciju klasificēšanai, labāk būtu jāizveidot treniņkopu, kura satur fotogrāfijas no vairākām šāda tipa maratonu attēlojošām kolekcijām, lai modelis no katras kopas iemācītos svarīgākas iezīmes; protams, arī vajadzētu daudz lielāku treniņkopu, lai tiktu ņemta vērā iespējamo objektu dažādība.

Lai izveidotu notikumu klasifikatoru, ir jāveic ilgs un sarežģīts analizēšanas process, lai noteiktu objektu apzīmējošas notikuma vietas, un objektu savstarpējo saistību.

IZMANTOTĀ LITERATŪRA UN AVOTI

1. Błażej Osipiński & Konrad Budek. What is reinforcement learning? The complete guide [tiešsaiste] – [atsauce 17.04.2022].
Pieejams: [What is reinforcement learning? The complete guide - deepsense.ai](https://deepsense.ai)
2. Ian Goodfellow, Yoshua Bengio & Aaron Courville. Deep Learning, 104. lpp. [tiešsaiste] – [atsauce 17.04.2022].
Pieejams: <https://www.deeplearningbook.org/contents/ml.html>
3. What Is Machine Learning: Definition, Types, Applications and Examples - Potentia Analytics [tiešsaiste] – [atsauce 17.04.2022].
Pieejams: [What Is Machine Learning: Definition, Types, Applications and Examples - Potentia Analytics \(potentiaco.com\)](https://potentiaco.com)
4. IBM Cloud Education. Unsupervised Learning [tiešsaiste] – [atsauce 17.04.2022].
Pieejams: [What is Unsupervised Learning? | IBM](https://www.ibm.com/cloud/learn/unsupervised-learning)
5. Unsupervised Learning: Algorithms and Examples [tiešsaiste] – [atsauce 17.04.2022].
Pieejams: [Unsupervised Machine Learning: Examples and Use Cases | AltexSoft](https://www.altexsoft.com/blog/unsupervised-machine-learning-examples-and-use-cases)
6. IBM Cloud Education. Supervised Learning [tiešsaiste] – [atsauce 17.04.2022].
Pieejams: [What is Supervised Learning? | IBM](https://www.ibm.com/cloud/learn/supervised-learning)
7. Labeled Data [tiešsaiste] – [atsauce 10.05.2022].
[What is Labeled Data? - Definition from Techopedia](https://www.techopedia.com/definition/35419/what-is-labeled-data)
8. Sunayana G. Domadia & Dr.Tanish Zaveri. Comparative Analysis of Unsupervised and Supervised Image Classification Techniques, 3. – 5. lpp. [tiešsaiste] – [atsauce 18.04.2022].
Pieejams: [401043.pdf \(bvmengineering.ac.in\)](https://www.bvmengineering.ac.in/401043.pdf)
9. IBM Cloud Education. Deep Learning [tiešsaiste] – [atsauce 11.05.2022].
Pieejams: [What is Deep Learning? | IBM](https://www.ibm.com/cloud/learn/deep-learning).
10. Jonathan Johnson. What's a Deep Neural Network? Deep Nets Explained [tiešsaiste] – [atsauce 11.05.2022].
Pieejams: [What's a Deep Neural Network? Deep Nets Explained – BMC Software | Blogs](https://www.bmc.com/blogs/whats-a-deep-neural-network-deep-nets-explained).
11. Ewan. What is Image Recognition? [tiešsaiste] – [atsauce 19.05.2022].
Pieejams: [Image Recognition : A Complete Guide - Deepomatic](https://deepomatic.com/blog/image-recognition-a-complete-guide)
12. Taha Anwar. Classification with Localization: Convert any Keras Classifier to a Detector [tiešsaiste] – [atsauce 22.04.2022].
Pieejams: [Classification with Localization: Convert any Keras Classifier to a Detector |](https://www.analyticsvidhya.com/blog/2022/04/classification-with-localization-convert-any-keras-classifier-to-a-detector/)

[LearnOpenCV #](#)

13. Mrinal Tyagi. Mathematical and practical implementation of various image segmentation techniques [tiešsaiste] – [atsauce 27.04.2022].
Pieejams: [Image Segmentation: Part 1. Mathematical and practical... | by Mrinal Tyagi | Towards Data Science](#)
14. Hmrishav Bandyopadhyay. Image Segmentation: Deep Learning vs Traditional [tiešsaiste] – [atsauce 26.04.2022].
Pieejams: [Image Segmentation: Deep Learning vs Traditional \[Guide\] \(v7labs.com\)](#)
15. Alexander Kirillov, et al. Panoptic Segmentation [tiešsaiste] – [atsauce 28.04.2022].
Pieejams: [1801.00868.pdf \(arxiv.org\)](#)
16. Kavish Sanghvi. Image Classification Techniques [tiešsaiste] – [atsauce 16.05.2022].
Pieejams: [Image Classification Techniques. Image classification refers to a... | by Kavish Sanghvi | Analytics Vidhya | Medium](#)
17. What is convolutional neural network? [tiešsaiste] – [atsauce 25.04.2022].
Pieejams: [What is convolutional neural network? - Definition from WhatIs.com \(techtarget.com\)](#)
18. Simran Bansari. Introduction to how CNNs Work [tiešsaiste] – [atsauce 25.04.2022].
Pieejams: [Introduction to how CNNs Work. Introduction to how CNNs work | by Simran Bansari | DataDrivenInvestor](#)
19. CNN | Introduction to Pooling Layer [tiešsaiste] – [atsauce 26.04.2022].
Pieejams: [CNN | Introduction to Pooling Layer - GeeksforGeeks](#)
20. Niall O' Mahony, et al. Deep Learning vs. Traditional Computer Vision [tiešsaiste] – [atsauce 14.05.2022].
Pieejams: [1910.13796.pdf \(arxiv.org\)](#)
21. Renu Khandelwal. Computer Vision — A journey from CNN to Mask R-CNN and YOLO -Part 1 [tiešsaiste] – [atsauce 29.04.2022].
Pieejams: [Computer Vision — A journey from CNN to Mask R-CNN and YOLO -Part 1 | by Renu Khandelwal | Towards Data Science](#)
22. “Fast R-CNN and Faster R-CNN” [tiešsaiste] – [atsauce 12.05.2022].
Pieejams: [“Fast R-CNN and Faster R-CNN” \(jhui.github.io\).](#)
23. Jatin Prakash. Non Maximum Suppression: Theory and Implementation in PyTorch [tiešsaiste] – [atsauce 12.05.2022].
Pieejams: [Non Maximum Suppression: Theory and Implementation in PyTorch \(learnopencv.com\)](#)
24. Aakarsh Yelisetty. Understanding Fast R-CNN and Faster R-CNN for Object Detection

- [tiešsaiste] – [atsauce 13.05.2022].
Pieejams: [Understanding Fast R-CNN and Faster R-CNN for Object Detection. | by Aakarsh Yelisetty | Towards Data Science](#)
25. Sambasivarao, K. Region of Interest Pooling [tiešsaiste] – [atsauce 13.05.2022].
Pieejams: [Region of Interest Pooling. A Technique which allowed a new... | by Sambasivarao. K | Towards Data Science](#)
26. Pulkit Sharma. A Step-by-Step Introduction to the Basic Object Detection Algorithms (Part 1) [tiešsaiste] – [atsauce 17.05.2022].
Pieejams: [Introduction to Object Detection Algorithms \(analyticsvidhya.com\)](#)
27. Elisha Odemakinde. Mask R-CNN: A Beginner's Guide [tiešsaiste] – [atsauce 15.05.2022].
Pieejams: [Mask R-CNN: A Beginner's Guide - viso.ai](#)
28. Grace Karimi. Introduction to YOLO Algorithm for Object Detection [tiešsaiste] – [atsauce 15.05.2022].
[Introduction to YOLO Algorithm for Object Detection | Engineering Education \(EngEd\) Program | Section](#)
29. Hmrishav Bandyopadhyay. YOLO: Real-Time Object Detection Explained [tiešsaiste] – [atsauce 15.05.2022].
Pieejams: [YOLO: Real-Time Object Detection Explained \(v7labs.com\)](#)
30. Rafal Kapela, Kevin McGuinness & Noel E. O'Connor. Real-time field sports scene classification using colour and frequency space decompositions [tiešsaiste] – [atsauce 28.05.2022].
Pieejams: [Real-time field sports scene classification using colour and frequency space decompositions | SpringerLink](#)
31. Shweta Pardeshi. CNN-LSTM Architecture and Image Captioning [tiešsaiste] – [atsauce 26.05.2022].
Pieejams: [CNN-LSTM Architecture and Image Captioning | by Shweta Pardeshi | Analytics Vidhya | Medium](#)
32. Andrey V. Savchenko & Evgeniy V. Miasnikov. Event Recognition Based on Classification of Generated Image Captions [tiešsaiste] – [atsauce 29.05.2022].
Pieejams: [Event Recognition Based on Classification of Generated Image Captions | SpringerLink](#)
33. Ojārs Millers. Latvijas čempionāts pagarinātajā distancē [tiešsaiste] – [atsauce 20.04.2022].
Pieejams: [2021-09-26 Latvijas čempionāts pagarinātajā distancē | Flickr](#)

34. Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo & Ross Girshick.
Detectron2 [tiešsaiste] – [atsauce 07.05.2022].
Pieejams: [facebookresearch/detectron2: Detectron2 is a platform for object detection, segmentation and other visual recognition tasks. \(github.com\)](https://github.com/facebookresearch/detectron2)
35. Waleed Abdulla. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow [tiešsaiste] – [atsauce 21.04.2022].
Pieejams: [matterport/Mask_RCNN: Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow \(github.com\)](https://github.com/matterport/Mask_RCNN)
36. Sergio Canu. Train Mask R-CNN for Image Segmentation (online free gpu) [tiešsaiste] – [atsauce 23.04.2022].
Pieejams: [Train Mask R-CNN for Image Segmentation \(online free gpu\) - Pysource](https://pysource.com/train-mask-r-cnn-for-image-segmentation-online-free-gpu/)
37. François Chollet. Transfer learning & fine-tuning [tiešsaiste] – [atsauce 08.05.2022].
Pieejams: [Transfer learning & fine-tuning \(keras.io\)](https://keras.io/transfer-learning/)
38. Piotr Skalski. Make Sense [tiešsaiste] – [atsauce 26.04.2022].
Pieejams: [SkalskiP/make-sense: Free to use online tool for labelling photos. https://makesense.ai \(github.com\)](https://github.com/SkalskiP/make-sense)
39. Ajitesh Kumar. Machine Learning – Training, Validation & Test Data Set [tiešsaiste] – [atsauce 10.05.2022].
Pieejams: [Machine Learning - Training, Validation & Test Data Set - Data Analytics \(vitalflux.com\)](https://vitalflux.com/machine-learning-training-validation-test-data-set/)
40. Kizito Nyuytiyimbij. Parameters and Hyperparameters in Machine Learning and Deep Learning [tiešsaiste] – [atsauce 10.05.2022].
Pieejams: [Parameters, Hyperparameters, Machine Learning | Towards Data Science](https://towardsdatascience.com/parameters-hyperparameters-machine-learning-towards-data-science/)
41. Choosing number of Steps per Epoch in Tensorflow [tiešsaiste] – [atsauce 05.05.2022].
Pieejams: [Choosing number of Steps per Epoch in Tensorflow - PyQuestions.com - 1001 questions for Python developers](https://pyquestions.com/1001-questions-for-python-developers/)
42. Mareks Gaļinovskis. Skolu Kauss [tiešsaiste] – [atsauce 16.05.2022].
Pieejams: [2022 04 27 Skolu Kauss, IZLASE, Foto: Mareks Gaļinovskis / FotoManLV | Flickr](https://www.flickr.com/photos/mareksgalinovskis/2022042700000000/)

PIELIKUMI

1. pielikums: Augsta līmeņa klasificēšanas funkcijas kods

```
def classifyImage(results, bigGroupMin, bigGroupMax):  
  
    """  
    Funkcija veic fotogrāfijas augsta līmeņa klasificēšanu,  
    balstoties uz instanču segmentēšanas modeļa atgriztajiem  
    rezultātiem.  
  
    /Params/  
    results:          Saraksts ar Mask-RCNN modeļa iegūtajiem  
                    rezultātiem ("class_ids", "scores" etc.)  
  
    bigGroupMin:     Subjektīvi definētais minimālais cilvēku skaits  
                    lielā cilvēku grupā. Cilvēku skaits, kas ir mazāks  
                    par šo vērtību, definē mazo cilvēku grupu  
                    ("Maza cilvēku grupa" tiek definēta, ja ir atrasti  
                    vismaz 3 cilvēki).  
  
    bigGroupMax:     Subjektīvi definētais maksimālais cilvēku skaits  
                    lielā cilvēku grupā. Ja cilvēku skaits ir lielāks  
                    par šo vērtību, tad tiks izmantota kategorija "cilvēku  
                    pūlis".  
  
    /Returns/  
    peopleGroupSize: Kategorijas veida "cilvēku grupas lielums" kategorija  
    location:        Kategorijas veida "atrašanās vieta" kategorija  
    facingPosition:  Kategorijas veida "cilvēku novietojums pret kameru"  
                    kategorija  
  
    """  
  
    # Vārdnīca, kura sasaista definētos kategoriju  
    # nosaukumus ar instanču segmentēšanas modeļa  
    # klašu identifikatoriem (class_ids).  
    classDict = {  
        "car":          1,  
        "person":      2,  
        "face":         3,  
        "podium":       4,  
        "tent":         5,  
        "flag banner": 6,  
        "banner wall": 7,  
        "forest":       8,  
        "grass":        9,  
        "fence banner": 10  
    }  
}
```

```

# Atrašanās vietu kategorijas un to apzīmējošie objekti
#     winnersPodiumArea:    banner wall, podium
#     parkingArea:         car
#     mainEventArea:       tent, fence banner, flag banner
#     lawn:                 grass
#     forestFromInside:    forest

# Vārdnīca definē noteikto objektu apstrādes secību.
# (class_id, priority)
# Lielāks skaitlis - zemāka prioritāte.
classPriorityDict = {
    7: 1, # "banner wall"
    4: 1, # "podium"
    1: 2, # "car"
    5: 3, # "tent"
    10: 3, # "fence banner"
    6: 3, # "flag banner"
    9: 4, # "grass"
    8: 5, # "forest"
    3: 6, # "face"
    2: 7 # "person"
}

# Izveido sarakstu ar pāriem, kur katrs pāris apzīmē konkrētu klasi
# vienu reizi, un atrastās klases objektu skaitu.
unique, counts = np.unique(results['class_ids'], return_counts=True)
result = np.column_stack((unique, counts))

resultList = result.tolist()

# Katram kortežam pievieno 3 elementu - prioritāti
resultWithPriorities = []
for object in resultList:
    class_id = object[0]
    object.append(classPriorityDict[class_id])
    resultWithPriorities.append(object)

# Sakārto kortežus pēc 3 elementa dilstošā secībā
from operator import itemgetter
resultWithPrioritiesSorted = sorted(resultWithPriorities,
key=itemgetter(2), reverse=True)

# Mainīgie, kuri apzīmē definētos fotogrāfiju
# kolekcijas veidus.
location = "in Unknown Location"
personGroupSize = "Zero people"
facingPosition = ""

# Mainīgie, kuri tiek izmantoti "facingPosition" vērtības
# noteikšanai un izņēmuma gadījumu apstrādei.
personCountInt = 0

```

```

faceCountInt = 0

# Cikls, kurā apstrādā katru atrasto klasi,
# ņemot vērā arī atrasto objektu skaitu.
for object in resultWithPrioritiesSorted:
    class_id = object[0]
    count    = object[1]

    # Kategorijas veida "cilvēku grupas lielums" piešķiršana.
    if class_id == classDict["person"]:
        personCountInt = count

        if count == 1:
            personGroupSize = "One person"
        elif count == 2:
            personGroupSize = "Two people"
        elif count >= 3 and count < bigGroupMin:
            personGroupSize = f"Small group of people"
        elif count >= bigGroupMin and count <= bigGroupMax:
            personGroupSize = f"Big group of people"
        elif count > bigGroupMax:
            personGroupSize = f"Crowd of people"

        continue

    # Kategorijas veida "cilvēku novietojums pret kameru" piešķiršana.
    if class_id == classDict["face"]:
        faceCountInt = count

        if count >= (int(personCountInt / 2)) and personCountInt < 11:
            facingPosition = "facing camera"
        elif count >= (int(personCountInt / 3)) and personCountInt >= 11:
            facingPosition = "facing camera"
        elif count >= (int(personCountInt / 4)) and personCountInt >= 20:
            facingPosition = "facing camera"
        elif personCountInt >= 25: # Gandrīz vai neredz sejas
            facingPosition = ""      # tāpēc neko neraksta, jo īsti nevar
pateikt

        else:
            facingPosition = "not facing camera"

        continue

    # Atrašanās vietas piešķiršana
    if class_id == classDict["forest"]:
        location = "in Forest"
    elif class_id == classDict["grass"]:
        location = "on Lawn"
    elif class_id == classDict["tent"] or class_id == classDict["fence
banner"] or \

```

```

        class_id == classDict["flag banner"] or class_id ==
classDict["car"] and count < 3:
            location = "in Main Event Area"
        elif class_id == classDict["car"] and count >= 3:
            location = "in Parking Area"
        elif class_id == classDict["podium"] or class_id == classDict["banner
wall"]]:
            location = "in Winners Podium Area"

# Ja neatrada nevienu seju bet atrada cilvēkus no 1 - 24,
# tad cilvēki ir vērsti pert kameru ar muguru.
if faceCountInt == 0 and (personCountInt < 25 and personCountInt > 0):
    facingPosition = "not facing camera"

# Ja neatrada nevienu cilvēku, tad atstāj mainīgo tukšu
if personCountInt == 0:
    facingPosition = ""

# Izņēmumgadījums: Seju skaits ir lielāks nekā atrasto cilvēku skaits
# Tādā gadījumā cilvēku skaitu nosaka seju skaits
if personCountInt < faceCountInt:
    if faceCountInt == 1:
        personGroupSize = f"One person"
    elif faceCountInt == 2:
        personGroupSize = f"Two people"
    elif faceCountInt >= 3 and faceCountInt < bigGroupMin:
        personGroupSize = f"Small group of people"
    elif faceCountInt >= bigGroupMin and faceCountInt <= bigGroupMax:
        personGroupSize = f"Big group of people"
    elif faceCountInt > bigGroupMax:
        personGroupSize = f"Crowd of people"

# Izprintē fotogrāfijas augsta līmeņa kategoriju
print("Classified as: ", personGroupSize, location, facingPosition)

return personGroupSize, location, facingPosition

```