

FLAI.AI – SPLETNA APLIKACIJA ZA SAMODEJNO KLASIFIKACIJO OBLAKA TOČK

dr. Klemen Čotar, Jernej Nejc Dougan

Flai d.o.o.

klemen.cotar@flai.ai, ORCID: <https://orcid.org/0000-0002-3439-0858>

nejc.dougan@flai.ai

DOI: https://doi.org/10.3986/9789610506683_23

UDK: 528.8.044.6:004.9

IZVLEČEK

Flai.ai – spletna aplikacija za samodejno klasifikacijo oblaka točk

Izdelali smo spletno aplikacijo za delo z lidarskimi podatki, katerega osredni del je postopek samodejne klasifikacije oblaka točk. Aplikacija uporabniku omogoča uvoz lastnih podatkov, njihov pregled ter pogajanje algoritmov, kot so klasifikacija, filtriranje in izdelava poljubnih modelov površja. Rezultate klasifikacij je možno tudi popraviti, uporabiti za ponovno učenje modelov klasifikacije in prenesti v izvorni obliki.

KLJUČNE BESEDE

lidarski podatki, klasifikacija, globoko učenje, oblaki točk, spletna aplikacija

ABSTRACT

Flai.ai – web application for automatic classification of point clouds

We have created a web application designed to work with lidar data, whose core functionality is the procedure for the automatic classification of point clouds. The application allows the user to import their own data, review them and apply algorithms such as classification, filtering and creation of custom terrain models. Classified results can be manually corrected, used to re-learn classification models and downloaded in the original format.

KEY WORDS

lidar data, classification, deep learning, point clouds, web application

1 Uvod

Čedalje obširnejša in pogostejša lidarska snemanja površja povečujejo tudi potrebo po zanesljivih samodejnih orodjih za samodejno klasifikacijo surovih oblakov točk. Ponudniki snemanj oziroma njihovi naročniki velikokrat uporabljajo polsamodejna orodja, ki za izdelavo končni izdelkov potrebujejo veliko prilagajanja in številne nadaljnje ročne popravke. Ti popravki so odvisni od zahtevnosti snemanega območja ter lastnosti objektov, ki jih želimo razpoznati iz podatkov med procesom klasifikacije. Najpogostejša razreda, ki jih na tak način ločujemo v podatkih, sta teren in stavbe, preostali objekti pa ostanejo v razredu vegetacije.

Za poenostavitev celotnega procesa smo razvili samodejni sistem, ki pohitri in poenostavi postopek klasifikacije, uporabniku sistema pa na enostaven način omogoča dodajanje novih klasifikacijskih razredov, ki jih želi označiti v podatkih. Algoritem klasifikacije temelji na principu konvolucijskih nevronskih mrež (LeCun s sodelavci 1989; Krizhevsky, Sutskever in Hinton 2012) z uporabo globokega učenja na podlagi množice ročno klasificiranih podatkov.

Sistem smo do zdaj uspešno prilagodili in uporabili za klasifikacijo obsežnejših preglednih skeniranj z malim številom zahtevanih razredov, podrobna snemanja koridorjev z večjim številom razredov ter snemanja z visoko gostoto točk za popis gozdnih sestojev. Celoten sistem klasifikacije je možno pogajati v oblaku preko spletne aplikacije ali na lastni strojni opremi.

2 Metode globokega učenja za oblake točk

Področje raziskovanja globokega učenja za klasifikacije in identifikacije objektov se je prvotno osredotočalo na slikovne podatkovne zbirke. V zadnjih nekaj letih pa se vedno več prispevkov tudi v domači literaturi ukvarja s samodejno obravnavo oblakov točk (Mongus, Triglav Čekada in Žalik 2013; Mongus in Horvat 2015; Mlekuž 2018; Kranjec, Triglav Čekada in Kobal 2021). Ti so lahko pridobljeni z detektorji, pritrjenimi na zračne (letalo, helikopter, letalniki), mobilne (avto, vlak) ali stacionarne platforme. Ne glede na način zajema se ti podatki razlikujejo od slikovnih formatov. Oblaki točk so množice podatkov, kjer prostorski odnosi med elementi niso določeni s podatkovno strukturo, ampak jih je treba izračunati. Obenem je njihova površinska gostota močno odvisna od načina zajema, oddaljenosti od objekta in prekrivanja med zajemi.

Lastnosti lidarskih podatkov so vodile v več različnih postopkov predpriprave samih vhodnih podatkov za lažjo uporabo znotraj modelov globokega učenja. Poznamo modele, ki oblak točk iz različnih perspektiv pogleda preslikajo v slikovni format (Qi s sodelavci 2017a; Zhao s sodelavci 2019), uporabijo točke za zapolnitev diskretnega trirazsežnostnega prostora (Graham, Engelcke in Van Der Maaten 2018; Meng s sodelavci 2019), izvedejo postopke primerjav med koordinatami in značilkami točk (Thomas s sodelavci 2019) ali pa uporabijo kombinacijo teh prijemov.

Med vsemi postopki predpriprave podatkov za strojno učenje v zadnjem času prevladujejo postopki, ki delujejo na izvirnih neobdelanih oblakih točk. Ti dajejo najboljše rezultate, vendar obenem zahtevajo največjo dodatno računsko moč za izračun prostorskih relacij med podatki. Glede na način uporabe vhodnih podatkov ločimo sklope: uporaba točk brez njihovih medsebojnih povezav (Qi s sodelavci 2017a; Qi s sodelavci 2017b), konvolucijski postopki (Li s sodelavci 2018; Thomas s sodelavci 2019), pogled na točke kot povezane točke modela grafa (Chen s sodelavci 2019; Wang s sodelavci 2019) ali pogled na točke z vidika hierarhične drevesne strukture (Zeng in Gevers 2018).

Jedro naše implementirane procesne verige globokega učenja temelji na konvolucijskih nevronskih mrežah za oblake točk (Qi s sodelavci 2017b; Thomas s sodelavci 2019; Zhao s sodelavci 2021), ki posnemajo delovanje operatorjev konvolucije za slikovne podatke. Konvolucijski operator je tako, namesto dvo-dimenzionalne pravokotne oblike, tri-dimenzionalna krogla. V njej so točke konvolucijskega jedra razporejene v dve koncentrični sferi, ki jima je dodana središčna točka. Vpliv vsake točke jedra upada z evklidsko oddaljenostjo, uteženo z izbrano funkcijo, ponavadi linearno, katere vpliv se izniči na izbra-

ni razdalji proč od njene lokacije. Efektivno obliko konvolucijskega operatorja tako določajo individualne uteži točk jedra, ki se določijo med postopkom učenja nevronske mreže. Na vsaki ravni nevronske mreže imamo več naborov uteži, katerih število se podvoji s prehodom na globljo raven.

Uporabljena nevronska mreža temelji na arhitekturi U-Net (Ronneberger, Fischer in Brox 2015) in jo sestavlja 6 vhodnih, 1 sredinski in 6 izhodnih ravni, ki delujejo po principu kodirnika in dekodirnika. Prostorska razdalja med točkami se povečuje proti sredini ter zmanjšuje proti izhodu, katerega točke so enake vhodu. Za vmesne ravni mreže ključne točke ustvarimo iz prvotnih, pri čemer med njimi želimo obdržati čim večjo medsebojno razdaljo. Tekom postopka konvolucije se konvolucijska jedra zasidrajo na ključne točke višjega nivoja, kamor se vpišejo tudi konvolucijski rezultati kot nove značilke, operacija konvolucije pa se izvede na značilkah nižjega nivoja. Prehodi iz sredinskega nivoja proti izhodu modela ne uporabljajo konvolucijskega operatorja, vendar le prenos in združevanje vrednosti značilk najbližjim točkam višje prostorske ločljivosti.

3 Postopek semantične klasifikacije

3.1 Predpriprava podatkov

Ne glede na način uporabe podatkovne zbirke, je le-tega potrebno pravilno pripraviti in iz njega izračunati podporne informacije za delovanje modela nevronske mreže. Prvi korak uvoza podatkov je izenačitev prostorske gostote podatkov po celotni zbirki, saj se le-ta lahko razlikuje zaradi različnih lastnosti opazovanih objektov ali prekrivanja več vzporednih zajemov. Z izenačitvijo gostot želimo različne zajeme medsebojno uravnotežiti in preprečiti, da bi se model naučil tudi gostote točk posameznega razreda ali podatkovne zbirke, saj ima vsaka edinstvene zahteve in lastnosti snemanja. Za izenačitev prostorskih gostot izvedemo podvzorčenje vhodnih podatkov tako, da celoten prostor razdelimo na enako velike kocke. Znotraj vsake kocke dopustimo največ eno točko, ki je v primeru večjega števila izbrana naključno. Prostornina kocke je prilagojena glede na velikost opazovanih objektov, najpogostejše gostote zračnih snemanj ter omejitve največjega števila točk, ki jih hkrati lahko uporabimo med učenjem nevronske mreže na grafični kartici. V primeru iskanja manjših objektov in podrobnejših skeniranj je ta parameter nastavljen na manjše vrednosti.

Tako podvzorčene podatkovne zbirke so večinoma še vedno občutno prevelike za hkratno obdelavo na grafični kartici, zato iz njih izrežemo pokončne valje podatkov (število valjev, ki jih obdelamo hkrati, je odvisno od tega, ali gre za postopek učenja ali projekcijo) s polmerom nekaj 10 metrov. V primeru skeniranja površja so podatki razporejeni ravninsko, kar zagotavlja predvidljive količine podatkov znotraj posameznega izreza. Takšno izrezovanje podatkov ima za posledico izgubo konteksta z okoljskimi izpuščenimi podatki. Tako je ravna betonska streha stavbe, ki je večja od izreza, skoraj nerazpoznavna v primerjavi z ravno betonsko površino na tleh. Objekta bi se potencialno razlikovala le v dodatkih, ki bi lahko stali nad njima.

Pred samo uporabo v nevronske mreži je iz vhodnih podatkov potrebno določiti še ključne točke, kamor se bodo zapisali rezultati operacij konvolucije. Njihovo število upada z naraščajočo globino znotraj modela. Optimalna medsebojna razdalja obenem narašča in je izračunana kot četrtnina polmera krogle konvolucijskega jedra. Največja velikost krogle je določena glede na velikost objektov, ki jih želimo razpoznati. Minimalna velikost krogle skupaj s podvzorčenjem na vhodnem nivoju določata kakšne detajle lahko zaznamo.

3.2 Učna zbirka in učenje modela

Trenutno uporabljamo učno zbirko 650 izsekov velikosti 120 krat 120 m (delček prikazuje slika 1), ki so bili pridobljeni iz različnih podatkovnih zbirk. Te so bile posnete na različnih geografskih območjih,

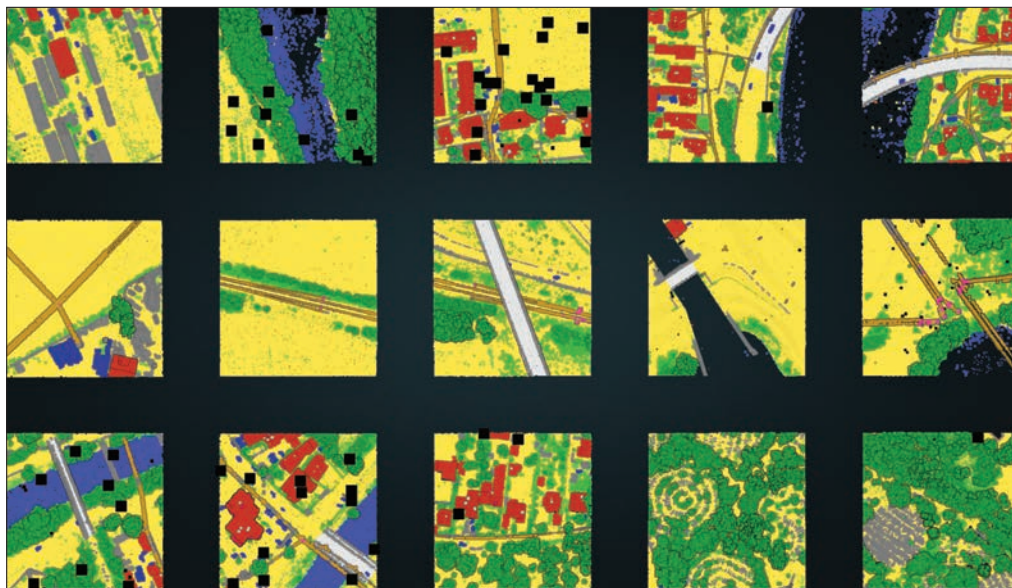
zajete z različnimi senzorji in tehnikami snemanja ter klasificirane za različne končne namene. Med njihovim izenačevanjem smo na enovito pripravljenih zbirkah naučili modele, jih prenesli na vse ostale ter razrede združili. Tako pridobljen skupek klasifikacij posameznih razredov še ni bil popoln in uporaben za nadaljnje učenje, kar smo rešili z natančnimi ročnimi popravki in dopolnitvami. Učno zbirko smo pripravili za razpoznavo 13 razredov: teren, vegetacija, strehe stavb, stene stavb, objekti na strehah, točke pod terenom, vodne površine, daljnovodi, daljnovodni stolpi, mostovi, šum visoko nad terenom in vozila.

Učenje globokega modela se izvaja v korakih, kjer med vsakim korakom algoritmu posamezen učni izsek pokažemo le enkrat, njihovo zaporedje pa vsakokrat naključno spremenimo. Sami izseki so nameroma večji od polmera izrezanega valja podatkov, saj s tem zagotovimo dodatno spremenljivost vhodnih podatkov s premikanjem središča valja znotraj meja, ki zagotavljajo popolno ploskovno pokritost valja s podatki. Dodatno spremenljivost zagotavljamo z naključnim vrtenjem valja okrog svoje osi, dodajanjem šuma koordinatam točk, napihovanjem in krčenjem izseka ter spreminjanjem lidarskih značilk. Med učenjem uporabljamo informacije o številu odbojev in njihovo intenziteto, ki jo z enakomerno porazdelitvijo spreminjamo za 30 %, saj je le-ta zelo odvisna od lastnosti snemanj.

Posamezen model pred uporabo učimo v 200 korakih, ki se izvedejo približno v enem dnevu. Ocenjevanje natančnosti naučenih modelov se v literaturi pogosto ocenjuje z metriko F1, ki je za posamezen razred definirana kot

$$F_1 = \frac{TP}{TP + 0,5 (FP + FN)},$$

kjer TP označuje število pravilno klasificiranih točk, seštevek vrednosti FP in FN pa podaja število nepravilno označenih in neoznačenih točk opazovanega razreda. Naučeni model na nevideni testni zbirki doseže povprečno vrednost metrike $F_1 = 0,87$ čez vse razrede. Za najbolj zastopane razrede, kot so vegetacija, teren in stavbe, znaša vrednost metrike več kot 0,95.



Slika 1: Pogled na izreze učne zbirke. Različno obarvane točke oblaka so vegetacija (zelena), stavbe (rdeča), objekti na strehah (svetlo modra), mostovi (bela), vozila (temno modra), teren (rumena), vodne površine (modra), daljnovodi (oker), nerazporejeno (siva) in šum (črna).

3.3 Aktivno učenje in grajenje učne zbirke

Trenutna učna zbirka je nastajala postopoma z združevanjem različnih zbirk v več korakih. Ob prejetju novega podatkovnega seta smo najprej izvedli učenje na takratni učni zbirki, naučeni model uporabili na novih podatkih in iz rezultatov projekcije določili tipe objektov, ki naučenemu modelu povzročajo probleme pri razpoznavi. To določanje je bilo izvedeno na dva načina: samodejno določanje ter ročno pregledovanje. Med samodejnim določanjem smo na podlagi rezultatov projekcij pregledali vse rezultate in določil območja, kjer je imelo samodejno določanje največjo nedoločenost odločitev, torej kjer je projekcija kot izhod vrnila zelo podobni oziroma enaki verjetnosti za pripis točke v dva ali več različnih razredov. Takih območij se s samodejnim iskanjem najde veliko in so si lahko zelo podobna, zato smo vse predloge še pregledali, izbrali le najbolj reprezentativne in unikatne izseke ter jih dodali k učni zbirki.

Takšen ponavljajoč proces, imenovan aktivno učenje, je predvsem zelo uporaben za dodajanje novih razredov v obstoječo učno zbirko. Uporabnik lahko preko spletne aplikacije tako v le nekaj učnih izsekih ročno označi točke novega razreda, izvede novo učenje modela ter uporabi naučen model na učni zbirki. Na podlagi rezultatov in verjetnosti projekcij sistem uporabniku prikaže rezultate klasifikacij novega razreda, za katere je najmanj prepričan in uporabnika pozove k ročnim popravkom. Tako lahko v več ponovitvah procesa dodamo nov željeni klasifikacijski razred in izboljšamo natančnost njegove projekcije.

3.4 Uporaba modela na neznanih podatkih

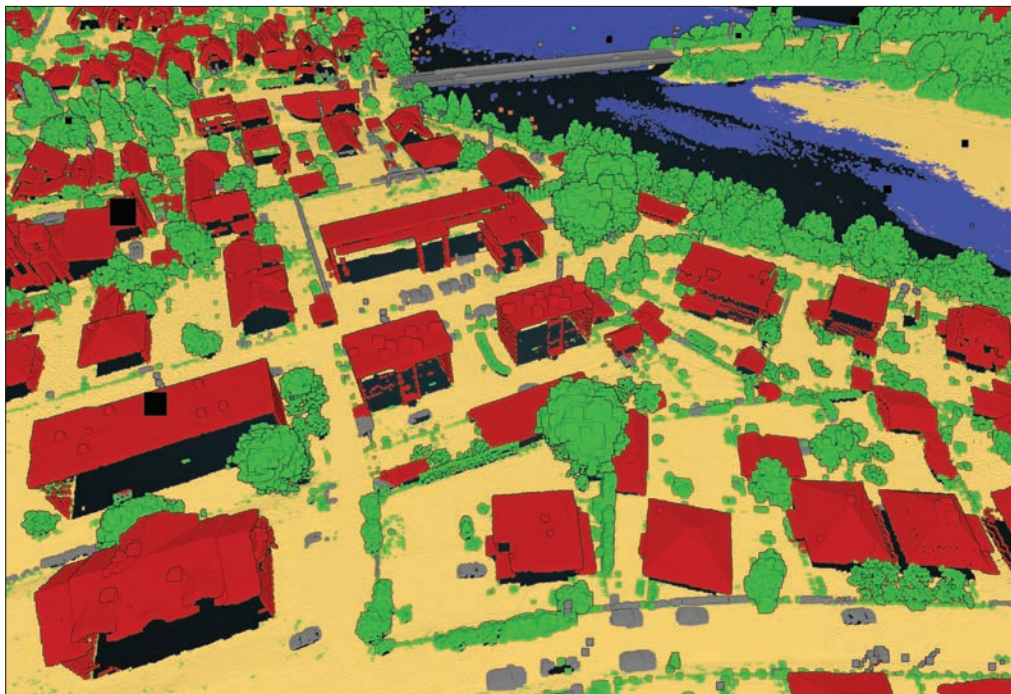
Cilj uporabe naučenega modela na izbrani podatkovni zbirki je klasifikacija vseh njegovih točk, kar pomeni, da moramo s čim ugodnejšim izrezovanjem prostorskih valjev vse podatke oblaka točk obdelati z naučenim modelom. V našem primeru smo centre valjev postavili v kvadratno mrežo, kjer je najdaljša dovoljena diagonala kvadrata mreže enaka dvakratniku polmera valja. Z manjšanjem razdalje med točkami podaljšamo čas procesiranja, vendar obenem povečamo prekrivanje med valji, s čimer dosežemo, da je vsaka točka klasificirana večkrat. V primeru več klasifikacijskih rezultatov na posamezno točko vzamemo tistega, ki mu je model pripisal večjo verjetnost klasifikacije. Z večkratnim pogledom na podatke tako rešujemo morebitne probleme izgube prostorskega konteksta na robovih valjev in na sredini prostorsko razsežnejših objektov. Med klasifikacijo zato uporabljamo razdaljo med točami mreže, ki je za četrtno manjša od največje vrednosti.

4 Rezultati in primerjava z ročnim delom

Do zdaj najizčrpnejši preizkus delovanja je bila klasifikacija nove neklasificirane zbirke, katere vzorci še niso bili vključeni v učno zbirko. Ročno dopolnjeni končni rezultati so bili kasneje dostavljeni naročniku snemanja. Celoten nabor podatkov je pokrival območje približno 3500 km² in bil za lažje delo razrezan v kvadratne izseke s površino 0,25 km². Naloga Flai je bila izvedba samodejne klasifikacije 8 razredov (stavbe, teren, mostovi, vodne površine, vegetacija, nerazvrščeno, visok šum in točke pod terenom, slika 2), ki bi čim bolj posnemala rezultate ročne klasifikacije. Dodatne razrede, ki jih naš model še prepozna, smo priključili najprimernejšemu od zahtevanih razredov.

Cilj samodejnega pristopa je pospešitev celotnega procesa od zajema podatkov do izdelave končnih izdelkov. Da bi preverili uporabnost samodejnega pristopa in zagotovili zahtevano kakovost končnega izdelka, je bil vsak samodejno obdelan izsek naknadno ročno pregledan in po potrebi popravljen.

Tako velika zbirka zaseda približno 9 TB diskovnega prostora, in še enkrat toliko za klasificirane podatke, saj med izvajanjem projekta ne želimo ničesar prepisovati. Zaradi velike količine podatkov ter razpolaganja z lastno strojno opremo smo vso obdelavo izvedli lokalno in ne na vzpostavljeno infrastrukturi v oblaku Amazon Web Service (AWS). Vse projekcije modelov so bile izvedene lokalno na dveh delovnih postajah, ki za pohitritev računanja uporabljata zmogljivejšo grafično kartico.



Slika 2: Rezultat samodejne klasifikacije s Flai, na katerem je zaznati manjše pomakljivosti. To sta pravilno razpoznavanje popolnoma ravnih površin (ločitev med vodo in terenom) ter neodločenost na stičišču objektov dveh ali več razredov.

4.1 Čas izvedbe

V nasprotju z ročno izvedbo klasifikacije čas obdelave s Flai ni odvisen od zahtevnosti površja ali števila razredov, temveč le od skupnega števila podatkovnih vnosov oblaka točk. Trenutna izvedba modela potrebuje približno 2 minuti za obdelavo 0,25 km² površja, kar skupaj zneso približno 20 dni računskega časa na delovni postaji z eno grafično kartico za površino 3500 km².

Po končani samodejni obdelavi je bil vsak izsek ročno pregledan in urejen. Ta operacija je zelo odvisna od pokritosti in rabe površine in je nanese približno 2 minuti za podeželske izseke, 20 minut za urbane izseke in 10 minut za mešane izseke. V obdelanem območju pripada večina izsekov podeželskim območjem, kar je skrajšalo čas nadzora kakovosti klasifikacije.

Za določitev pohitritve s samodejnim pristopom in primerjavo časovne zahtevnosti nadzora so bili iz vsakega tipa izseka popolnoma ročno klasificirani štirje izseki. Ročno delo seveda ne pomeni ročnega določevanja čisto vsake točke oblaka, saj so operatorju v uporabljenem programskem okolju na voljo številna vnaprej pripravljena preprosta orodja, ki podobno kot Flai delež dela opravijo samodejno. Ročni postopek je trajal približno 15 minut za podeželske izseke, 45 minut za mešane izseke in 90 minut za urbane izseke, kar predstavlja precejšno pohitritev procesa z našo izvedbo samodejne klasifikacije.

4.2 Natančnost klasifikacij

Natančnost ali podobnost med dvema naboroma podatkov je mogoče izračunati z uporabo množice metrik, ki se vse osredotočajo na poseben način primerjave dveh nizov. Izbira je odvisna od

informacije, ki jo želimo pridobiti z metriko. V našem primeru si želimo čim manj ročnega prerazporejanja točk med razredi za doseg želene končne natančnosti. Uporabljena metrika relativne zanesljivost oziroma Jaccardov koeficient (Jaccard 1912) je izračunana kot

$$R_{\text{razred}} = 100 \frac{N_{\text{zac}} - N_{\text{odst}}}{N_{\text{zac}} + N_{\text{dod}}},$$

kjer R_{razred} označuje zanesljivost preiskovanega razreda, ki je odvisna od števila lidarskih točk tega razreda pred popravkom N_{zac} , števila točk, ki so bile dodane razredu N_{dod} , in števila točk, ki so bile odstranjene iz razreda N_{odst} . Tipične zelene vrednosti metrike so nad 90 % za vse razrede, razen za razreda teren in nerazvrščeno, ki se običajno ne preverja, saj vsebuje vse neprepoznavne oziroma nerelevantne objekte. Zaželeno zanesljivost razpoznavne terena je nad 99 %, saj je to primarni izdelek tako obsežnih skeniranj. Za našo interno kontrolo kakovosti izdelka Flai smo v začetnih implementacijah želeli doseči zanesljivost nad 90 % za najbolj zastopane razrede, kot so teren, vegetacija in stavbe.

Da bi ocenili natančnost samodejne klasifikacije, smo iz celotne zbirke izbrali šest območij s površino 1 km² z različnimi konfiguracijami in tipi terena, kjer prevladujejo urbane regije, ki običajno zahtevajo največ ročnega dela in vsebujejo večino razredov. Rezultati, predstavljeni v preglednici 1, vsebujejo primerjavo med samodejno klasifikacijo Flai ter končnimi oddanimi klasificiranimi izdelki istega območja. Pri pregledu rezultatov je treba upoštevati tudi število točk v posameznem razredu, ki tu ni podano, saj le nekaj napačno razporejenih točk potencialno maloštevilčnih razredov, kot so šum nad terenom, točke pod terenom in mostovi, močno spremeni številsko oceno zanesljivosti.

Preglednica 1: Zanesljivost samodejne klasifikacije Flai za posamezni preiskovani razred v primerjavi s končnimi, ročno popravljenimi izdelki.

razredi	območje 1	območje 2	območje 3	območje 4	območje 5	območje 6
nerazvrščeno	46 %	44 %	33 %	40 %	38 %	55 %
teren	96 %	92 %	95 %	94 %	95 %	96 %
vegetacija	92 %	87 %	95 %	86 %	86 %	90 %
stavbe	93 %	95 %	92 %	93 %	95 %	94 %
točke pod terenom	90 %	15 %	57 %	15 %	28 %	43 %
vodne površine	/	78 %	/	23 %	/	/
mostovi	45 %	67 %	7 %	54 %	23 %	/
šum nad terenom	69 %	67 %	12 %	45 %	33 %	12 %

5 Spletna aplikacija

5.1 Uvoz in nadzor podatkov

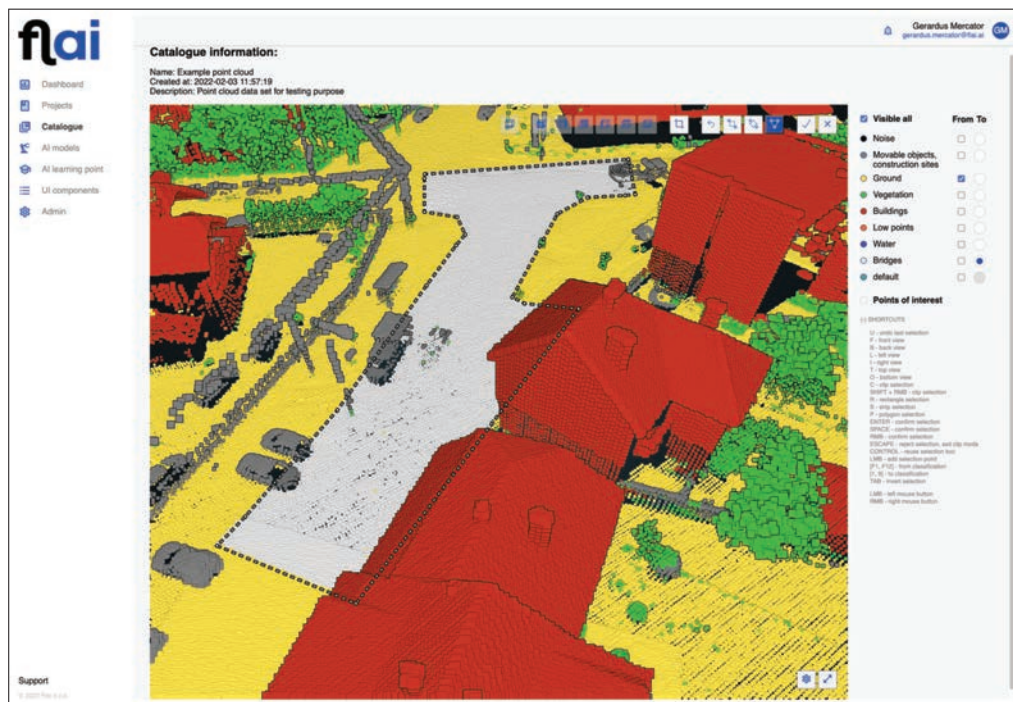
Prvi korak v uporabi spletne aplikacije je uvoz podatkov. Oblak točk je možno naložiti v obliki ene same LAS datoteke s hkratnim uvozom več datotek, ki so združene v arhivskem formatu, ali direktno preko podane povezave do podatkov na AWS oblaki storitvi Amazon Simple Storage Service (S3). Ob uvozu podatkov se ti prenesejo na strežnik in pripravijo za prikaz in nadaljnjo obdelavo. Uporabniki, ki so del iste organizacije, si med sabo lahko izmenjujejo uvožene podatkovne zbirke in podeljujejo pravice za nadzor nad njihovim dostopom.

5.2 Prikaz podatkovne zbirke

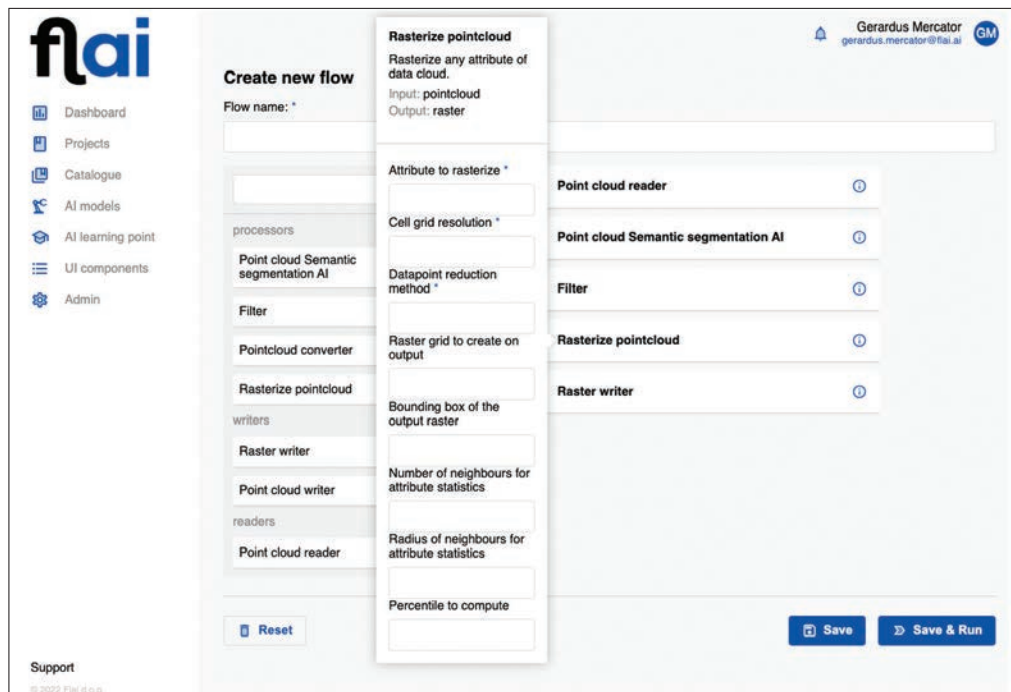
Eden pomembnejših orodij spletne aplikacije je prikazovalnik uvoženih podatkov, v katerem lahko poleg prikaza podatke tudi urejamo. S posebnimi namenskimi orodji lahko poljubno število razredov klasifikacij spremenimo v nov izbran razred. Točke, na katerih želimo izvesti preklasifikacijo, izberemo z orodji risanja pravokotnika, traku ali poljubnega n-kotnika (slika 3). Zahtevnejša opravila lahko izvedemo tudi v več korakih. Najprej s poljubno obliko izrežemo iz zbirke le delček podatkov in v naslednjem koraku s poljubno obliko določimo točke na katerih se bo popravek izvedel. Tako popravljene podatke lahko po končanem delu prenesemo z aplikacije v enaki obliki, kot so bili uvoženi.

5.3 Izvedba procesne verige

Ob uvozu podatkov se nam odklene možnost njihove obdelave z orodji, ki so na razpolago v aplikaciji. Uporabniku sta omogočeni poljubna izbira in razvrstitev orodij za izvedbo željene naloge. Slika 4 prikazuje, kako bi z dodajanjem posameznih orodij na naši zbirki pognali klasifikacijo, kjer bi iz rezultatov izvzeli le zelene točke, recimo teren, iz njih izdelali slikovni model terena in ga izvozili kot novo slikovno podatkovno zbirko. S klikom na posamezno orodje se nam odprejo dodatne nastavitve, s katerimi pridemo do zelenega rezultata. Izdelano procesno verigo je moč shraniti za kasnejšo ponovno uporabo na isti ali drugi podatkovni zbirki.



Slika 3: Ponazoritev uporabe prikazovalnika oblakov točk, vzpostavljen kot del spletne aplikacije, kjer z uporabo orodja za risanje poligonov (prekinjena črta) želimo spremeniti le točke terena v razred mostu označen s svetlo sivo barvo.



Slika 4: Gradniki procesne verige, potrebni za izvedbo klasifikacije terena in izdelavo slikovnega modela terena. Vsakemu izmed uporabljenih orodij lahko spremenimo nastavitve.

6 Sklep in nadaljnje delo

Samodejni postopki omogočajo hitro obdelavo velikih količin podatkov in razvrščanje v večje število kategorij, kot bi bilo ekonomično z ročnim označevanjem. To odpira tudi povsem nove možnosti uporabe podatkov, ki pred tem niso bile možne oziroma so bile stroškovno neupravičene. Prav tako na primeru izvedbe vsedržavnih zajemov podatkov, kjer so končni uporabniki iz najrazličnejših strok, ki imajo različne zahteve in opredelitve klasifikacije istih razredov, lahko s samodejnimi metodami obdelave omogočimo, da uporabniki izberejo najprimernejši končni način klasifikacije.

Od samega namena klasifikacije je odvisna tudi potreba po doseženi pravilnosti klasifikacije. Za nekatere aplikacije trenutni algoritem že več kot presega zahteve, druge za dosego potrebujejo še dodatno ročno preverjanje. Z vsako nadgradnjo se ta potreba zmanjšuje. Končni rezultat je kompromis med hitrostjo izvedbe in zahtevano pravilnostjo klasifikacije.

Najpomembnejši doprinos k kvaliteti končnih rezultatov je vsebinsko raznolika, vendar poenoteno pripravljena učna zbirka, ki jo stalno dopolnjujemo z novimi izseki in razredi. Poleg njenega nadgrajevanja načrtujemo implementacijo algoritma za detekcijo objektov znotraj oblaka točk ter vpejavo dodatnih algoritmov za filtriranje in naknadno obdelavo rezultatov. Le-ti bi odpravili trenutne pomanjkljivosti pri razpoznavi razredov točk, ki ležijo na stičiščih dveh ali več objektov različnih razredov. Takšen postopek bo vrnil estetsko lepše rezultate brez preskakovanj med klasifikacijskimi razredi na njihovih stičiščih.

7 Viri in literatura

- Chen, C., Li, G., Xu, R., Chen, T., Wang, M., Lin, L. 2019: Clusternet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis. IEEE/CVF conference on computer vision and pattern recognition. DOI: <https://doi.org/10.1109/CVPR.2019.00513>
- Graham, B., Engelcke, M., Van Der Maaten, L. 2018: 3d semantic segmentation with submanifold sparse convolutional networks. IEEE conference on computer vision and pattern recognition. DOI: <https://doi.org/10.1109/CVPR.2018.00961>
- Jaccard, P. 1912: The distribution of the flora in the alpine zone. *New phytologist* 11-2. DOI: <https://doi.org/10.1111/j.1469-8137.1912.tb05611.x>
- Kranjec, N., Triglav Čekada, M., Kobal, M., 2021. Napovedovanje drevesnih vrst iz geometrije in intenzitete oblaka aerolaserskih točk vrhov drevesnih krošenj. *Geodetski vestnik* 65-2. DOI: <https://doi.org/10.15292/geodetski-vestnik.2021.02.234-259>
- Krizhevsky, A., Sutskever, I., Hinton, G. E. 2012: Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*. DOI: <https://doi.org/10.1145/3065386>
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. 1989: Backpropagation applied to handwritten zip code recognition. *Neural computation*. DOI: <https://doi.org/10.1162/neco.1989.1.4.541>
- Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B. 2018: Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems* 31.
- Meng, H. Y., Gao, L., Lai, Y. K., Manocha, D. 2019: VV-net: Voxel vae net with group convolutions for point cloud segmentation. IEEE/CVF International Conference on Computer Vision. DOI: <https://doi.org/10.1109/ICCV.2019.00859>
- Mlekuž, D. 2018. Vzreja modelov višin iz lidarskih podatkov. *Pokrajina v visoki ločljivosti*. DOI: <https://doi.org/10.3986/9789610501138>
- Mongus, D., Horvat, D. 2015: gLidar – Napredno orodje za obdelavo lidarskih podatkov. *Geodetski vestnik* 59-1.
- Mongus, D., Triglav, M., Žalik, B. 2013. Analiza samodejne metode za generiranje digitalnih modelov reliefa iz podatkov lidar na območju Slovenije. *Geodetski vestnik* 57-2. DOI: <https://doi.org/10.15292/geodetski-vestnik.2013.02.045-259>
- Qi, C. R., Su, H., Mo, K., Guibas, L. J. 2017a: Pointnet: Deep learning on point sets for 3d classification and segmentation. IEEE conference on computer vision and pattern recognition. DOI: <https://doi.org/10.1109/CVPR.2017.16>
- Qi, C. R., Yi, L., Su, H., Guibas, L. J. 2017b: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems* 30.
- Ronneberger, O., Fischer, P., Brox, T. 2015: U-Net: Convolutional Networks for Biomedical Image Segmentation.
- Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., Guibas, L. J. 2019: KPConv: Flexible and deformable convolution for point clouds. IEEE/CVF International Conference on Computer Vision. DOI: <https://doi.org/10.1109/ICCV.2019.00651>
- Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., Solomon, J. M. 2019: Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics* 38-5. DOI: <http://dx.doi.org/10.1145/3326362>
- Zeng, W., Gevers, T. 2018: 3dcontextnet: Kd tree guided hierarchical learning of point clouds using local and global contextual cues. *European Conference on Computer Vision (ECCV) Workshops*. DOI: http://dx.doi.org/10.1007/978-3-030-11015-4_24
- Zhao, H., Jiang, L., Fu, C. W., Jia, J. 2019: Pointweb: Enhancing local neighborhood features for point cloud processing. IEEE/CVF conference on computer vision and pattern recognition. DOI: <https://doi.org/10.1109/CVPR.2019.00571>
- Zhao, H., Jiang, L., Jia, J., Torr, P. H., Koltun, V. 2021: Point transformer. IEEE/CVF International Conference on Computer Vision.