Tehnički izveštaj

Uporedjivanje metoda za semantičku segmentaciju slika nad slikama utakmica kriketa

Student:
Milica Aleksić 1642
Dušan Igić

Mentor:
prof.dr.Aleksandar Milosavljević

Sadržaj

Uvod

U okviru projekta implementirane su mreze UNet, DeepLabv3+ i SegNet. Demonstrirana je njihova primena za semanticku segmentaciju na setu podataka slika sa utakmica kriketa, preuzetog sa kaggle-a [1]. Slike iz dataset-a sadzre 9 semantickih klasa. Modeli su trenirani fiksnom stopom ucenja, a zatim i stopom ucenja koja opada sa porastopm broja epoha. Early stopping mehanizam integrisan je u sistem treniranja. Nakon treninga, modeli su evaluirani standardnim metrikama: mean IoU, mean accuracy, mean precision, mean recall i mean f1 score. Dobijene metrike iskoriscene su za uporedjivanje modela.

Teorijska osnova

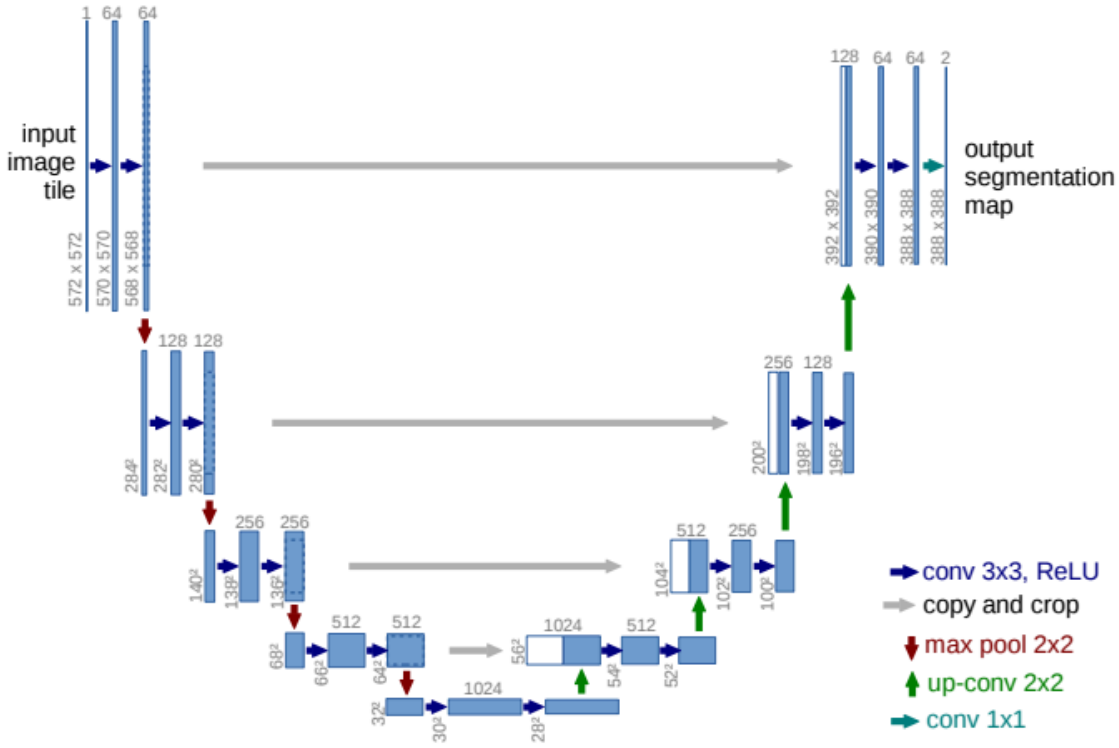
UNet

UNet je konvoluciona neuronska mreža razvijena za segmentaciju biomedicinskih slika, od strane Univerziteta u Frajburgu. Zasnovana na potpuno konvolucionoj neuronskoj mreži, čija je arhitektura modifikovana i proširena kako bi radila sa manjim brojem slika za obuku i davala preciznije segmentacije. Segmentacija slike veličine 512 × 512 piksela traje manje od sekunde na modernom GPU-u koristeći UNet arhitekturu. UNet je takođe primenjen u difuzionim modelima za iterativno uklanjanje šuma sa slika, što je osnova mnogih modernih modela za generisanje slika, kao što su DALL-E, Midjourney i Stable Diffusion. Omogućava preciznu segmentaciju sa relativno malim brojem slika za obuku, čineći je korisnom u mnogim područjima biomedicine i nauke o materijalima. Karakterišu je skip konekcije tj. prespojne veze medju ekvivalentinm slojevima u enkoder-dekoder arhitekturi.

Arhitektura

UNet se sastoji od kontraktivnog puta (enkoder) i ekspanzivnog puta (dekoder), što joj daje karakterističan u-oblik (slika x). Kontraktivni put je konvoluciona mreža koja se sastoji od ponovljene primene konvolucija, gde svaku konvoluciju prati ReLU (Rectified Linear Unit) aktivaciona funkcija i operacija maksimalnog grupisanja (max pooling). Tokom kontrakcije, prostorne informacije se smanjuju dok se informacije o karakteristikama povećavaju.

-



Slika x

Enkoder mreža

Enkoder mreža deluje kao ekstraktor karakteristika i uči apstraktnu reprezentaciju ulazne slike kroz sekvencu enkoder blokova. Svaki enkoder blok se sastoji od dve 3x3 konvolucije, gde svaku konvoluciju prati ReLU aktivaciona funkcija. Nakon toga sledi 2x2 maksimalno grupisanje, gde se prostorne dimenzije (visina i širina) redukuju za polovinu.

Dekoder mreža

Dekoder mreža se koristi za generisanje maske segmentacije. Dekoder blok počinje sa 2x2 transponovanom konvolucijom. Zatim se konkatenira sa odgovarajućim mapama karakteristika iz enkoder bloka putem prespojnih veza (skip connections). Ove prespojne veze pružaju dodatne informacije koje pomažu dekoderu da generiše bolje semantičke karakteristike. Nakon toga, koriste se dve 3x3 konvolucije, gde svaku konvoluciju prati ReLU aktivaciona funkcija. Izlaz poslednjeg dekoder bloka prolazi kroz 1x1 konvoluciju sa sigmoidnom aktivacijom, koja daje masku segmentacije koja predstavlja piksel-po-piksel klasifikaciju.

Prespojne veze(skip connections)

Prespojne veze pružaju dodatne informacije dekoderu koje pomažu u generisanju boljih semantičkih karakteristika. Takođe, omogućavaju bolji protok gradijenata tokom povratnog širenja, što pomaže mreži da bolje nauči reprezentaciju.

Most

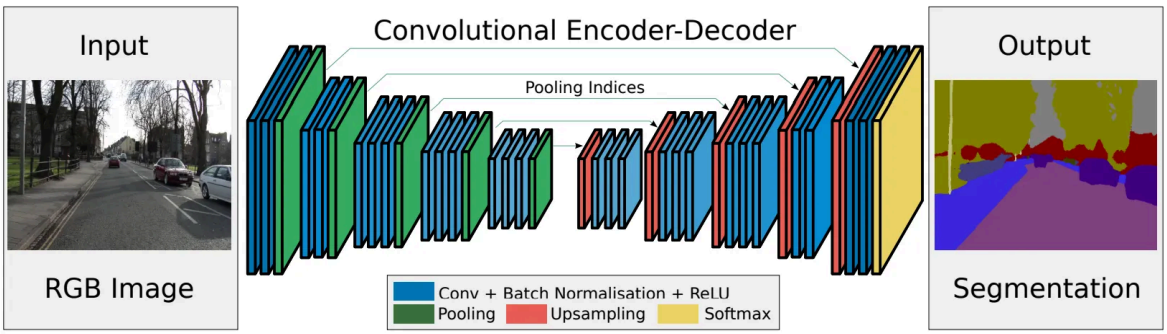
Most povezuje enkoder i dekoder mrežu i omogućava protok informacija. Sastoji se od dve 3x3 konvolucije, gde svaku konvoluciju prati ReLU aktivaciona funkcija. [2]

SegNet

SegNet je duboka konvoluciona neuronska mreža, implementirana za semantičku segmentaciju u računarskom vidu. Razvijena je od strane Kembridž Univerziteta. Karakteriše je enkoder-dekoder arhitektura sa specifičnim pristupom up-sampling, koristeći indekse maksimalnog grupisanje, što omogućava efikasno očuvanje prostorne informacije i detaljnu rekonstrukciju slika. Oba tehnika čini SegNet izuzetno memorijski-efikasnom i brzom za obradu većih skupova podataka. Prednost u odnosu na UNet je brzina i memorijska efikasnost, a mana je gore očuvanje prostornih informacija, koje UNet-u omogućavaju skip konekcije.

Arhitektura

SegNet se sastoji od enkoderskog i dekoderskog dela, zajedno sa završnim slojem za klasifikaciju, Ova arhitektura je bazirana na 13 konvolucionih slojeva VGG'16 modela, pri čemu se ključni indeksi maksimalnog grupisanje pamte tokom enkodiranja radi kasnijeg up-samplinga u dekoderskoj fazi (slika 2).



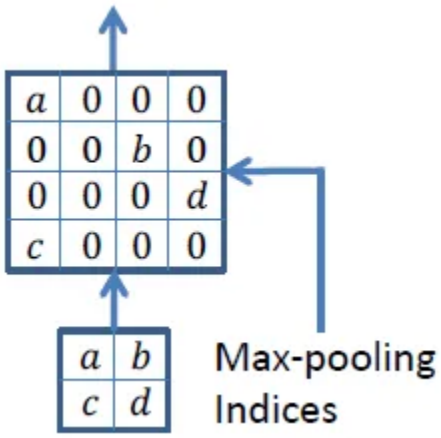
Slika 2.

Enkoder mreža

Enkoderski deo SegNet-a funkcioniše kao ekstrator karakteristika. Koristi sekvence konvolucionih slojeva i maksimalnog grupisanje za smanjenje prostornih dimenzija slike i ekstrakciju visokog nivoa fičera. Svaki konvolucioni sloj praćen je ReLU aktivacionom funkcijom radi učenja nefunkcionalnih reprezentacija slike.

Dekoder mreža

Dekoderski deo SegNet-a koristi sačuvane indekse maksimalnog grupisanje iz enkoderske faze za precizno up-samplovanje karakteristika i generisanje konačne maske segmentacije. Ova faza uključuje konkatenaciju karakteristika sa odgovarajućih enkoderskih slojeva, što omogućava bolje razumevanje i interpretaciju konteksta (slika 3).[2]



Slika 3.

DeepLabv3+

DeepLav3+ predstavlja arhitekturu za semantičku segmentaciju koja unapredjuje svoje prethodnika, posedno DeepLabv3, uvodeći efikasan dekoder modul zasnovan na atrous separabilnim kovolucijama radi poboljšanja rezultat segmentacije. Kombinuje Atrous Spatial Pyramid Pooling iz DeepLabv1 i enkoder-dekoder arhitekturu iz DeepLabv2. Ova kombinacija omogućava bolje hvatanje višeskalnih kontekstualnih informacija i precizniju segmentaciju.

Arhitektura

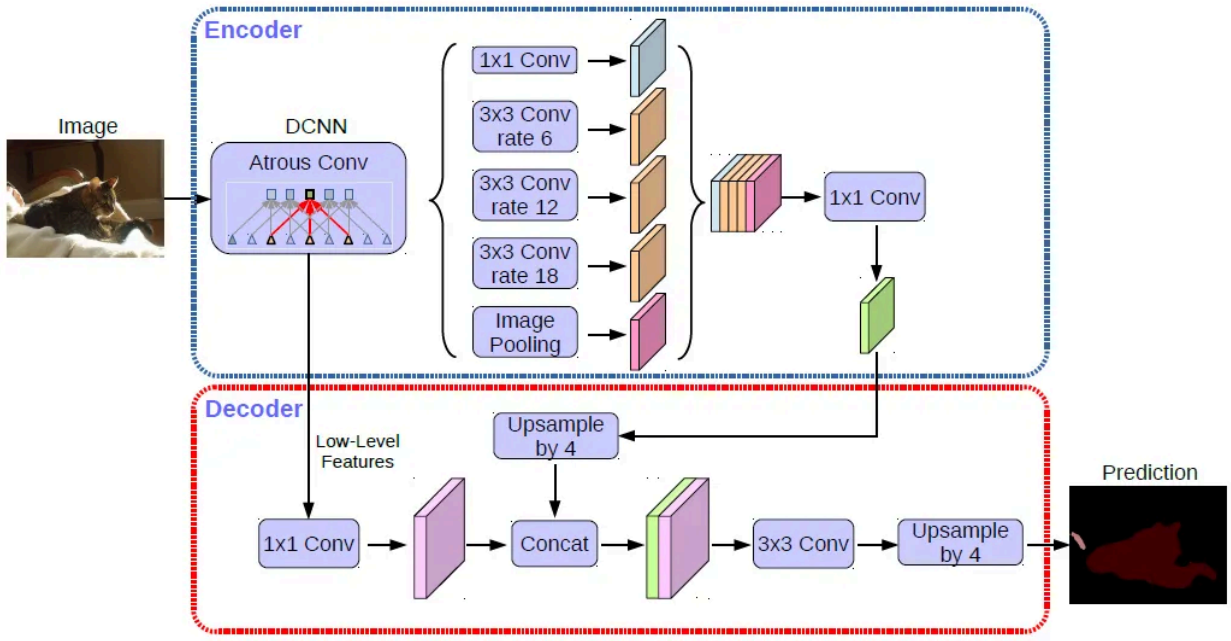
DeepLabv3+ arhitektura kombinuje karakteristike enkoder-dekoder strukture sa naprednim konvolucionim tehnikama. Glavni enkoder ove arhitekture je modifikovana verzija

ResNet 50, koja duboko ekstrutkuje razlicite nivoe karakteristika iz ulazne slike. Paralelno sa glavnim enkoderom, koristi se druga instance ResNet50 za hvatanje detaljnih karakteristika niskog nivoa.

Nakon sto prodje kroz enkoder, ulazi u Atrous Spatial Pyramid Pooling (ASPP) modul. Atrous Spatial Pyramid Pooling koristi atrous konvolucije sa razlicitim stopama dilacije kako bi uhvatio informacije slika na vise skala. Dodatno, primenjuje se globalna konvolucija na nivou slike kako bi se dobila sira kontekstualna informacija.

U dekoderu se uzorkuju karakteristike iz ASPP i konkateniraju sa karakteristikama niskog nivoa. Pre konkatenacije, primenjuje se 1x1 konvolucija, radi smanjenja broja kanala i balansiranja znacajnosti razlicitih nivoa karakteristika. Nakon toga, primenjuju se 3x3 konvolucije za dalje refiniranje karakteristika.

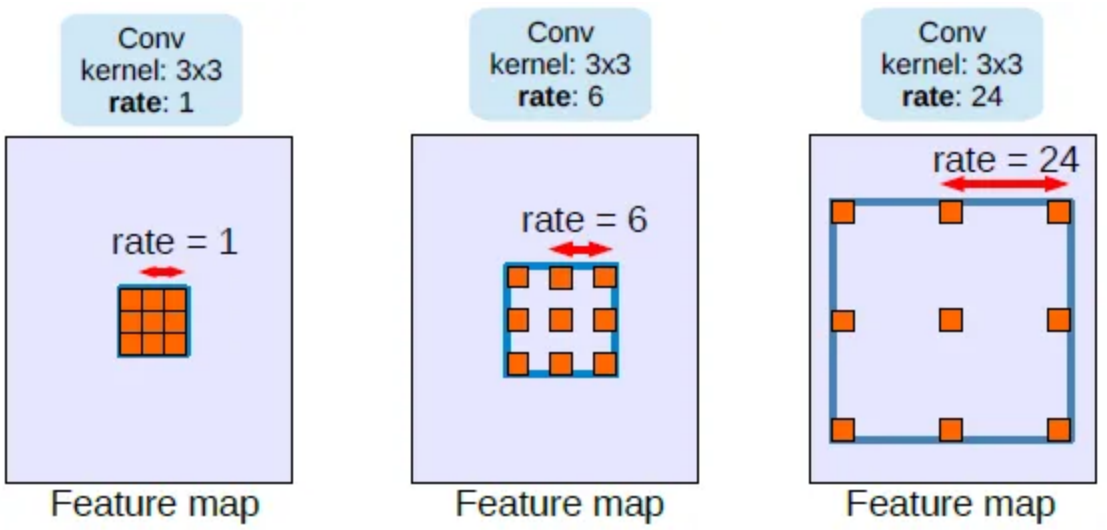
Na kraju, izlazne karakteristike se upsample-uju bilinearnom interpolacijom i prolaze kroz 1x1 konvoluciju kako bi se generisala krajnja segmentaciona maska.(slika 4.)



Slika 4.

Atrous (Dilated) konvolucija

Dilatirane konvolucije uvode dodatan parametar u konvolucione slojeve nazvan dilatacioni faktor 'r'. Dilatacioni faktor kontroliše razmak između tačaka jezgra (slika 5). Kontrolom parametra faktora može se proizvoljno kontrolisati receptivna polja sloja konvolucije. Receptivno polje je definisano kao veličina regiona ulazne karakteristike mape koja proizvodi svaki izlazni element. Ovo omogućava konvolucionom filteru da pregleda veće oblasti ulazne mape bez smanjenja prostorne rezolucije ili povećanja veličine jezgra . [4]



Slika 5.

Tehnologije

Kao tehnologije za izradu projekta upotrebljenji su programski jezik Python, Google Colab platforma i PyTorch okruženje. Python je pogodan za razvoj dubokog učenja zbog svoje jednostavnosti, čitljivosti i dostupnim bibliotekama. Biblioteke poput NumPy olakšavaju manipulaciji i transformaciju podataka. Pillow i OpenCV su biblioteke dostupne za rad sa slikama i konverziju u NumPy nizove.
Google Colab je platforma zasnovana na cloud-u, implementiana zarad pojednostavljanja i ubrzanja razvoja dubokog učenja. Pruža:

- besplatan pristup visokoperformantnom hardveru poput GPU-a i TPU-a, što natno ubrzava proces,
- interaktivno okruženje, čime se olakšava proces vizuelizacije
- Integrisanje sa Google Drive-om, olakšavajući skladištenje i preuzimanje podataka i čuvanje težina modela.

PyTorch je besplatno okruzenje za kreiranje modela dubokog ucenja. Ima dobru podrsku za rad sa GPU-ovima i koristi reverse-mode auto-differentiation. Reverse-mode auto-differentiation se svodi na snimanje rezultata zavrsenih operacija i vracanja unazad kako bi se izracunali gradijenti. Podrzava dinamicke racunarske grafove, omogucavajuci promenu ponasanje mreze u runtime-u. Osnovni tip podataka je tensor, slicni visedimenzionalnom nizu, gde se cuvaju podaci. Pruza module za rad sa:

- Neuronskim mrezama torch.nn - izgradnja, treniranje modela
- Problemima racunarskog vida torchvision - unapred trenirani modeli, standardni skupovi podatka i transformacija slika
- Rad sa korisnickim klasama torch.utils.data - rad sa skupovima podataka putem klasa Dataset i DataLoader

Arhitektura sistema

Sistem se sastoji iz 8 logickih celina:

- Ucitavanje podataka
- Podela dataseta na trening, validacioni i test deo
- Kreiranje klase Dataset
- Kreiranje klasa modela
- Implementiranje pomocnih funkcija za treniranje
- Implementiranje pomocnih funkcija za testiranje
- Odabir CPU/GPU
- Treniranja
- Testiranja

U okviru ucitavanja podataka,vrsi se instaliranje biblioteke potrebne ucitavanje podataka diretko sa kaggle platforme. Nakon toga, preuzimaju se podaci sa kaggle platforme i ispituje se njihova struktura. Dostupna su dva foldera classes i images. U okviru classes foldera nalazi se json fajl koji sadrzi informacije o broju klasa, boji, u heksa formatu, kojom su oznaceni pikseli te klase, nazivu klase, zastupljenosti klase itd. Na osnovu ucitanog json fajla, kreira se lookup hes mapa, koja za kljuc ima redni broj klase, a za vrednost boju u rgb formatu. Ucitavaju se i slike iz foldera images. Slika su dostupne u png formatu kao i odgovarajuce maske. Naziv maske ima dodataka “__fuse” u imenu. Kreira se lista imena dostupnih slika. Implementira se i funkcija, kojom se maska iz rgba formata prevodi u matricu dimenzija 9 x m x n, gde su m in dimenzije slike, a 9 oznacava broj klasa. Piksel ce imati 1 na i-tom indeksu nulte dimenzije ukoliko pripada i-toj klasi.

Nakon upoznavanja sa dataset-om, lista imena slika se deli na tri dela: trening, validacioni i test deo, u razmeri 60:20:20.

Kreira se klasa CustomDataset potrebna za rad sa skupom podataka zadatog problema. Ona nasledjuje torch.utils.data.Dataset. Omogucava jednostavno ucitavanje, transformaciju i augmentaciju slika i njihovih odgovarajucih maski. Neophodno je definisati metode klase poput funkcije inicijalizacije __init__, funkciju koja vraca duzinu dataset-a __len__ i funkciju kojom se pristupa elementu __getitem__. Funkcija __init__ za parametre ima listu slika koje se koriste images_path, broj klasa za segmentaciju n_class i opcioni parametar transform kojim se navode zeljene transformacije nad slikama. U okviru nje, ucitavaju se sve slike i odgovarajuce maske iz liste slika. Skaliraju se na dimenzije 512 x 512 x3 za slike i 512x512x9 za maske. Ubacuju se i horizontalno flipovane verzije slika i odgovarajucih maski . Funkcija __len__ nema parametre i vraca ukupan broj skupa u podatku. Funkcija __getitem__ za parametar ima index slike i vraca

odgovarajucu sliku i masku na osnovu indeksa. Ukoliko su definisane transformacije, primenjuje ih nad slikom.

Svaki model kreiran je kao klasa, koja sadrzi metode inicijalizacije __init__ i kretanja podataka __forward__. U okviru klase inicijalizacije kao argument se prosledjue broj klasa n_class, a u funckiju __forward__ input promenjliva.

Klasa Unet enkapsulira istoimeni model. Tokom inicijalizacije kreiraju se objekti klasa Conv_block, Encoder_block i Decoder_block. Conv_block klasa definise blok konvolucione mreze koji se sastoji od dva sloja konvolucije, svaki pracen sloje batch normalizacije i ReLU aktivacionom funkcijom. Encoder_block klasa definise blok enkodera u U-Net arhitekturi, koji se sastoji od jednoh Conv_block-a i max pooling sloja. Decoder_block koristi transponovanu konvoluciju za povecanje dimenzionalnosti i spaja sa skip konekcijama, nakon cega primenjuje kovoluciju kroz Conv_block.

Klasa SegNet enkapsulira istoimeni model. Tokom inicijalizacije kreiraju se objekti klasa Encoder_block_segnet i Decoder_block_segnet. Conv_block_segnet klasa definise blok konvolucione mreze koji se sastoji konvolucije, pracene slojem batch normalizacije i ReLU aktivacionom funkcijom. Encoder_block_segnet klasa definise blok enkodera u SegNet arhitekturi, koji se sastoji od vise slojeva Conv_block_segnet-a, u zavisnosti od dubine bloka i pratecih max pooling slojeva. Forward funkcija ove klase vraca vrednosti i indekse maksimalnih vrednosti u okviru predela maksimizacije. Decoder_block_segnet koristi max unpool sloj i vise conv_block_segnet-a pri inicijalizaciji. MaxUnpool sloju se prosledjuju indeksi dobijeni prilikom max pooling-a.

Klasa DeepLabv3Plus se sastoji od 5 dela: backbone, low_level_features, assp, conv1x1 bloka, conv3x3 bloka i klasifikatora na kraju. Za backbone se koristi ResNet-50 model klase prethodno obucen do sloja layer3, tj 3.residualnog sloja. Njegovi izlazi se prosledjuju na ulazze ASSP modula, takodje implementiranog kao klasa. Zatim se dobijene vrednosti upsampluju sa faktorom 4. Istovreme se pocetni ulazi prosledjuju low_level_feature objektu, koji je implementiran kao ResNet50 sa prekidom na sloju layer1. Dobijeni izlazi se prosledjuju bloku conv1x1 koji je objekat tipa Atrous_Convolution. Rezultat se konkaternira sa upsample-ovanim vrednostima i prosledjuje se bloku conv3x3 koji predstavlja sekvencu konvolucije, batch normalizacije i relu aktivacione funkcije. Rezultat se upsample-uje faktorom 4 i prosledjuje klasifikatoru koji je predstavlja jedan sloj konvolucije. Atrous_convolution je klasa koja definise atrous konovlucij sa batch normalizacijom i ReLU aktivacionom funkcijom. ASSP klasa integrise karakteristike na vise skala koriscenjem atrous konvolucije i adaptivnim average pooling-om. Dilataciona stopa atrous konvolucija su redom: 1,6,12 i 18.

U implementiranje pomocnih funkcija za treniranje spada train funkcija sa fiksnom stopom ucenje, train funkcija koja dozvoljava opadavanje stope ucenja i funkcija za racunanje metrika. U okviru train funkcije kreiraju se dataload-eri za trening i validacioni set. Tada se prosledjuje kao transformacija normalizacija dataset-a na osnovu izracunate srednje vrednosti i standardne devijacije originalnih slika i horizontalno preslikanih, po svakog RGB kanalu. Za optimizator se koristi adam i kao loss funckija bira se CrossEntropyLoss. Kao parametri Adam optimizatora prosledjuju se parametri modela i stopa ucenja. Model, stopa ucenja i broj epoha se prosledjuju kao parametri train funckije. Na nivou epohe vrsi se treniranje, gde se za podatke data_x na nivou batch-a odredjuje predikcija y. Odredjuje se gubitak izmedju dobijenih vrednosti i labeliranih, back propagira gradijent i azurira model. Odredjuje se metrike, pozivom kreirane

funkcije, pruzajuci uvid u performanse na trening setu. Takode, se na kraju svake epohe model prebacuje u evaluacioni mod i racunaju performanse na valicadionim setu. Implementiran je i early stopping mehanizam, koji prati validacioni mIoU. Ako validacioni mIoU spadne vremenom tokom unapred zadatog broja epoha, trening se prekida. Tezine najboljeg modela se cuvaju u vidu matrice, a po okoncanju procesa treniranja, u okviru fajla na Google Drive-u. Funckija sa promenljivom stopom ucenja koristi StepLR learning rate scheduler dostupan u PyTorch-u, koji nakon step_size epoha, skalira learnign rate parametrom gamma. Funkcija za racunanje metrika racuna srednju vrednost tacnosti, preciznosti, odziva, f1 skor-a i IoU. Kao parametri prosledjene su dobijena mask i ground_trough maska, pri cemu je nad njima primenjena arg.max funckije koja smanjuje dimenzionalnost matrica tako sto se svaki vektor koji predstavlja klasu piksela menja indeksom jedinice ili najvece vrednoisti u vektoru.

U implementiranje pomocnih funkcija za testiranje smatra se implementiranje funkcije test, funkcije koja konvertuje tensor u numpy niz radi prikaza i funkcije za vizuelizaciju slika. Test funkcija je slicna validacionom delu trening funkcije. Funkcija za vizuelizaciju prikazuje do 10 slika. Za svaku sliku se prikazuje rgb slika, ground truth mask i prediktovana maska.

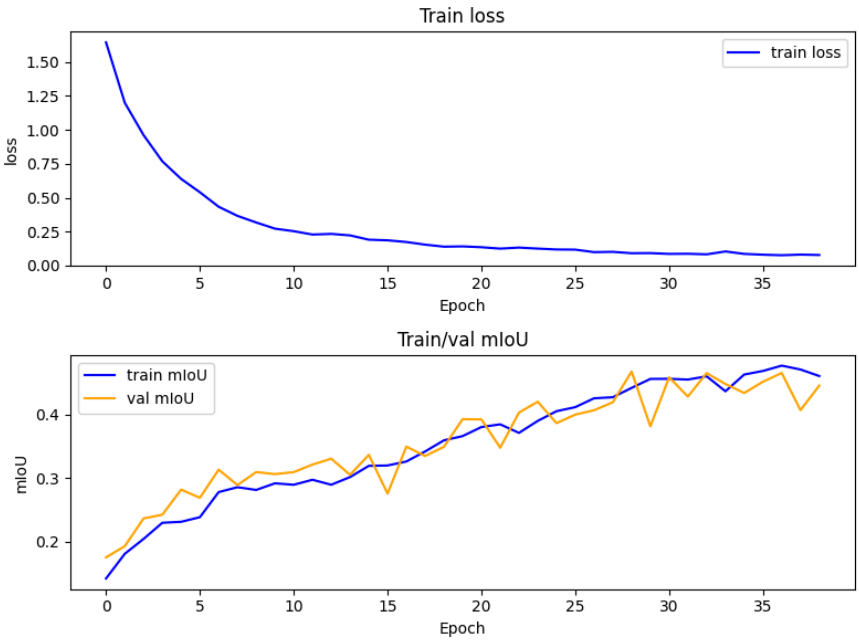
Nakon implementiranja svih funkcija podesava se uredjaj za izvrsenje trening i testiranja. Ukoliko je dostupan GPU opredeljuje se za njega. Nakon toga kreira se po dva modela UNeta, DeepLabv3Plusa i SegNeta, sa fiksnim learning rate-ovima 1e-3 i 1e-4. Na osnovu dobijenog validacionog mIoU, trenirani su modeli sa 50 epoha i boljom fiksnom stopom ucenja. Nakon toga modeli su trenirani sa promenljivom stopom ucenja na 50 epoha, koja opada posle svake 10. epohe za 10%. Modeli su zatim testirani na test setu i uporedjene su vrednosti.

Dataset

Dataset se sastoji od 298 slika sa utakmica kriketa. Slike su dimenzija 1080x1920x3 . Postoji 8 klasa Batsman, Bowler, Wicket Keeper, Fieder, Ball, Umpire, Wicket, Ground i Background objekta na slici. Uz slike dostupne su i maske u RGBA formatu. Zastupljenost klasa je redom 18.5%, 8.8%, 8.8%, 12.2%, 9.6%, 9.1%, 13.3%, 14.2% i 5.6%.

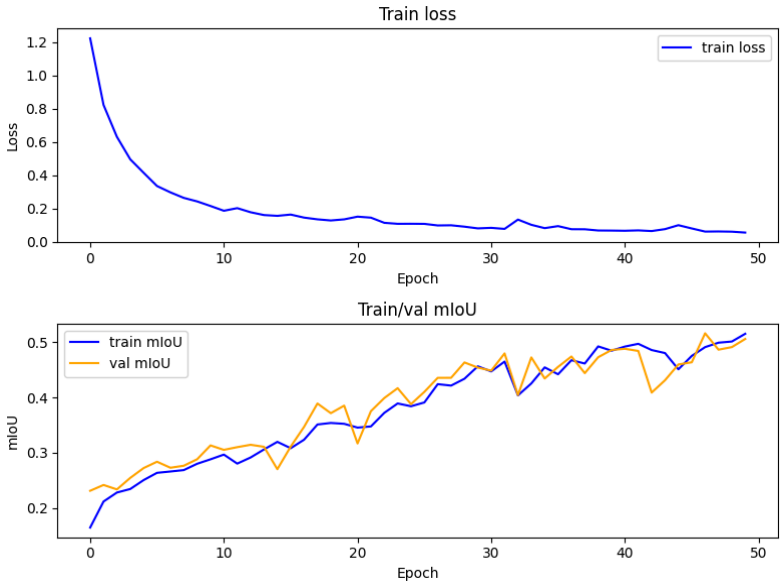
Rezultati

Model Unet , prilikom treninga sa 5 epoha, daje bolje rezultate sa learning rate-om 1e-4. Pri treniranju sa 50 epoha i fiksnim learning rate-om 1e-4 najveci mIoU postize u 29.epohi i iznosi 46.74%. Loss i mIoU tokom treninga prikazani su na slici 6, sa koje se moze uociti da loss funkcija treninga opada u skladu sa dobrim learning rate-om.



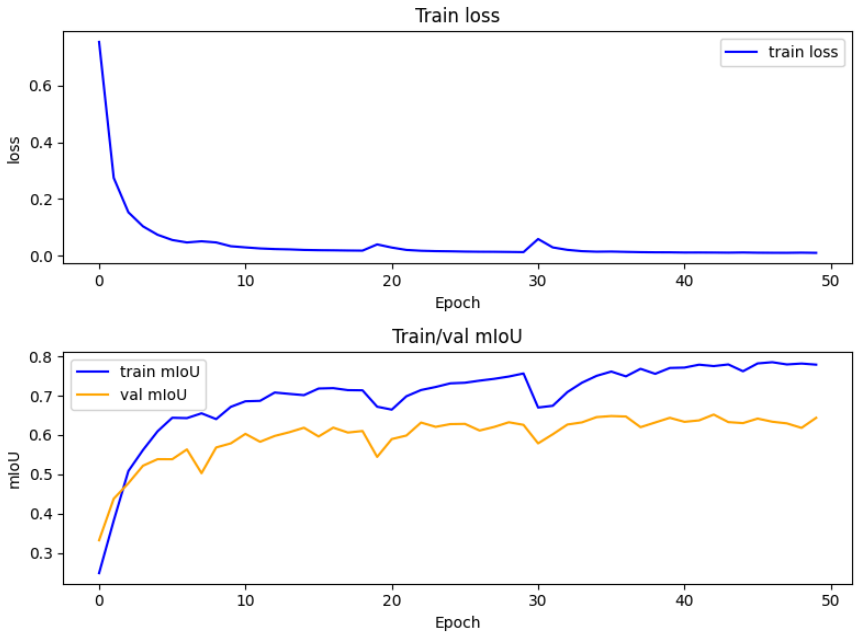
Slika 6.

Prilikom treniranja promenljivom stopom ucenja, model se trenira svih 50 epoha,ali najveci mIoU posize u 47.epohi i iznosi 51.61%. Na slici 7 prikazane su metrike modela prilikom treniranja.



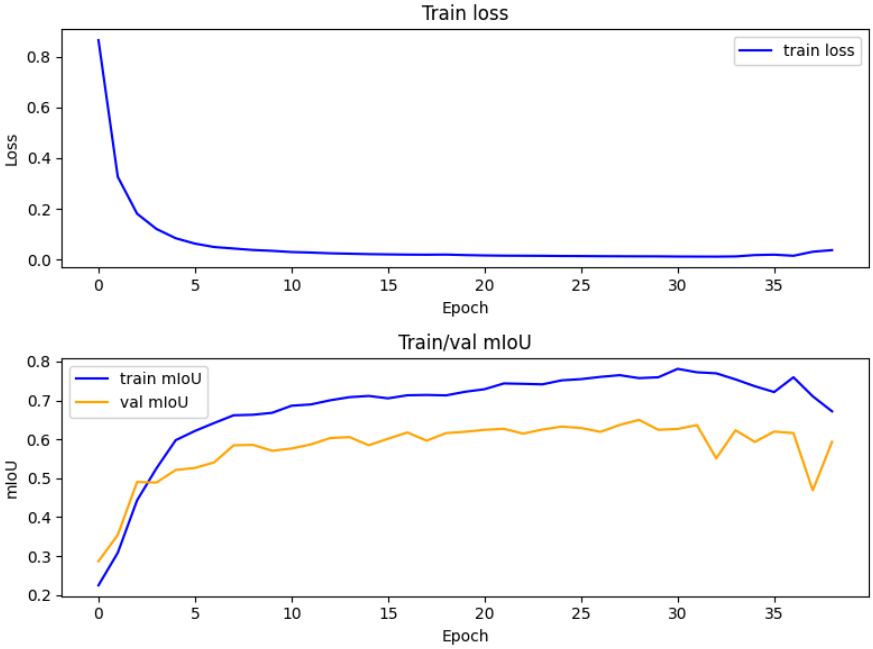
Slika 7.

Model DeepLabv3+ , prilikom treninga sa 5 epoha, daje bolje rezultate sa learning rate-om 1e-4. Pri treniranju sa 50 epoha i fiksnim learning rate-om 1e-4 najveci mIoU postize u 43.epohi i iznosi 65.19%. Loss i mIoU tokom treninga prikazani su na slici 8. Sa slike se moze proceniti da bi bilo pogodno probati trening sa vise epoha, ili u ovom slucaju, kako je early stopping prekinuo trening povecati broj neophodnih epoha.



Slika 8.

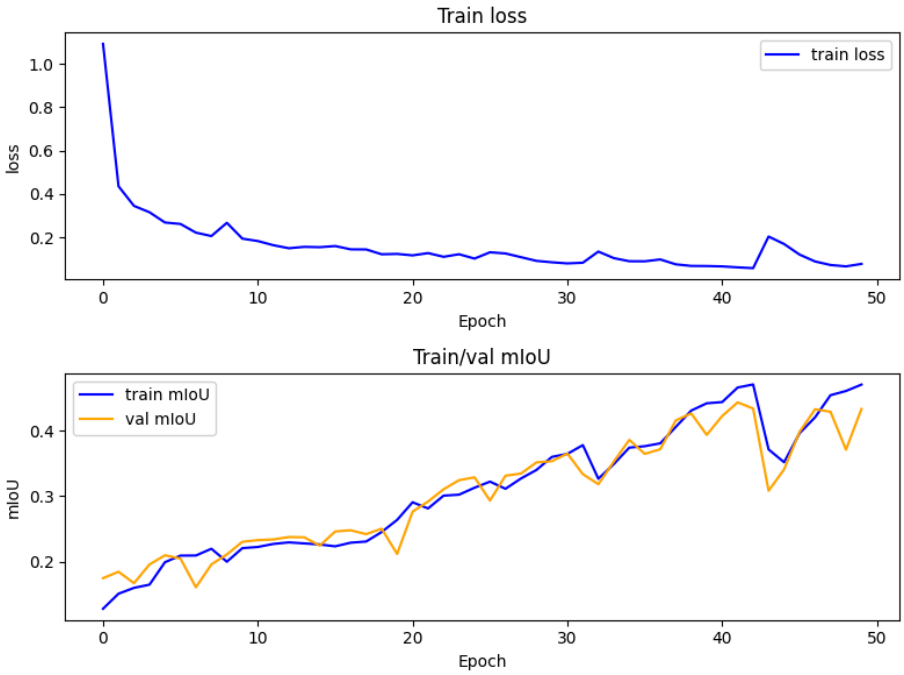
Prilikom treniranja sa promenljivim learning rate-om, koji opada od 1e-4, model najveci mIoU na validacionom dataset-u postize u 29.epohi i iznosi 65%. Perfonsa prilikom treninga prikazane su na slici 9.



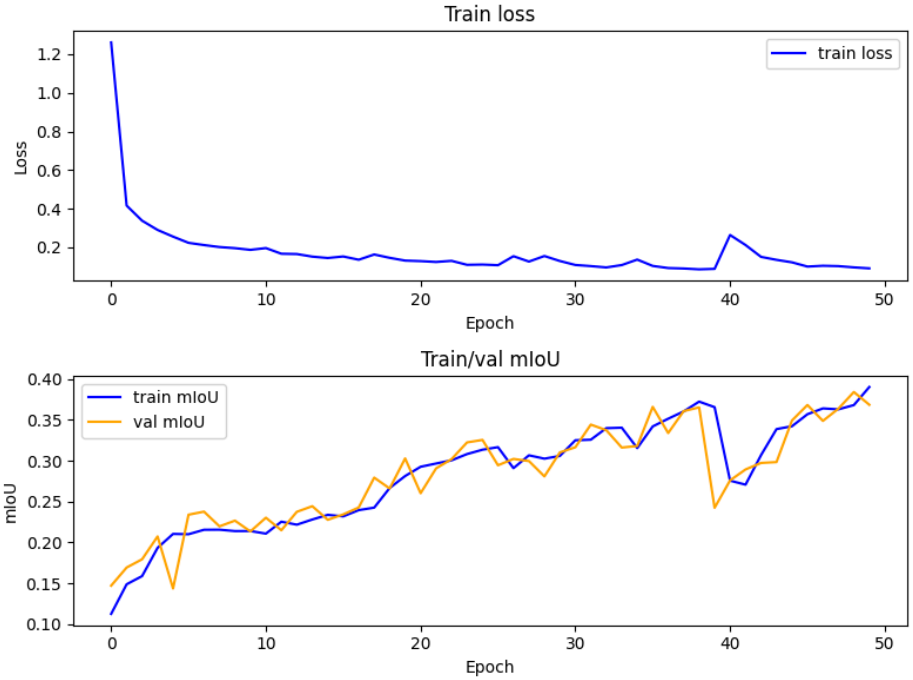
Slika 9.

Model SegNet , prilikom treninga sa 5 epoha, daje bolje rezultate sa learning rate-om 1e-4. Pri treniranju sa 50 epoha i fiksnim learning rate-om 1e-4 najveci mIoU postize u 42.epohi i iznosi 43.03%. Loss i mIoU tokom treninga prikazani su na slici 10. Prilikom treniranja sa

promenljivim learning rate-om najbolji mIoU ima u 40.epohi iznosi 38.4%. Na slici 11 su prikazane loss i mIoU tokom treninga.



Slika 10.



Slika 11.

U tabeli 1 su prikazane dobijene metrike na test skupu, modela treniranih sa fiksnim learning rate-om.

	mIoU	mAccuracy	mPrecision	mRecall	mF1 score
Unet	39.18%	99.35%	48.12%	45.22%	45%
DeepLabv3Plus	56.32%	99.77%	67.66%	60.48%	61.97%
SegNet	36.69%	99.13%	45.81%	45.37%	43.11%

Tabela 1.

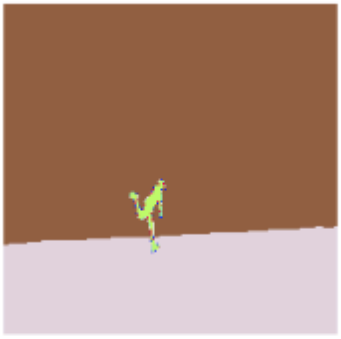
U tabeli 2 su prikazane dobijene metrike na test skupu, modela treniranih sa promenljivim learning rate-om, koji se skalira na 10 epoha sa 0,1.

	mIoU	mAccuracy	mPrecision	mRecall	mF1 score
Unet	41.7%	99.4%	50.53%	47.32%	46.93%
DeepLabv3Plus	56.63%	99.74%	67.12%	61.06%	62.37%
SegNet	32.77%	99.05%	42.73%	40.16%	38.01%

Tabela 2.

Neki od rezultata segmentacije su vizualizovani na slici 12. Slike pripadaju redom rezultatima modela Unet, DeepLabv3Plus i SegNet pri treningu sa fiksim i promenljivim learning rate-om. Na slici 11. se nalaze originalna slika i ground truth maska.

Original Image

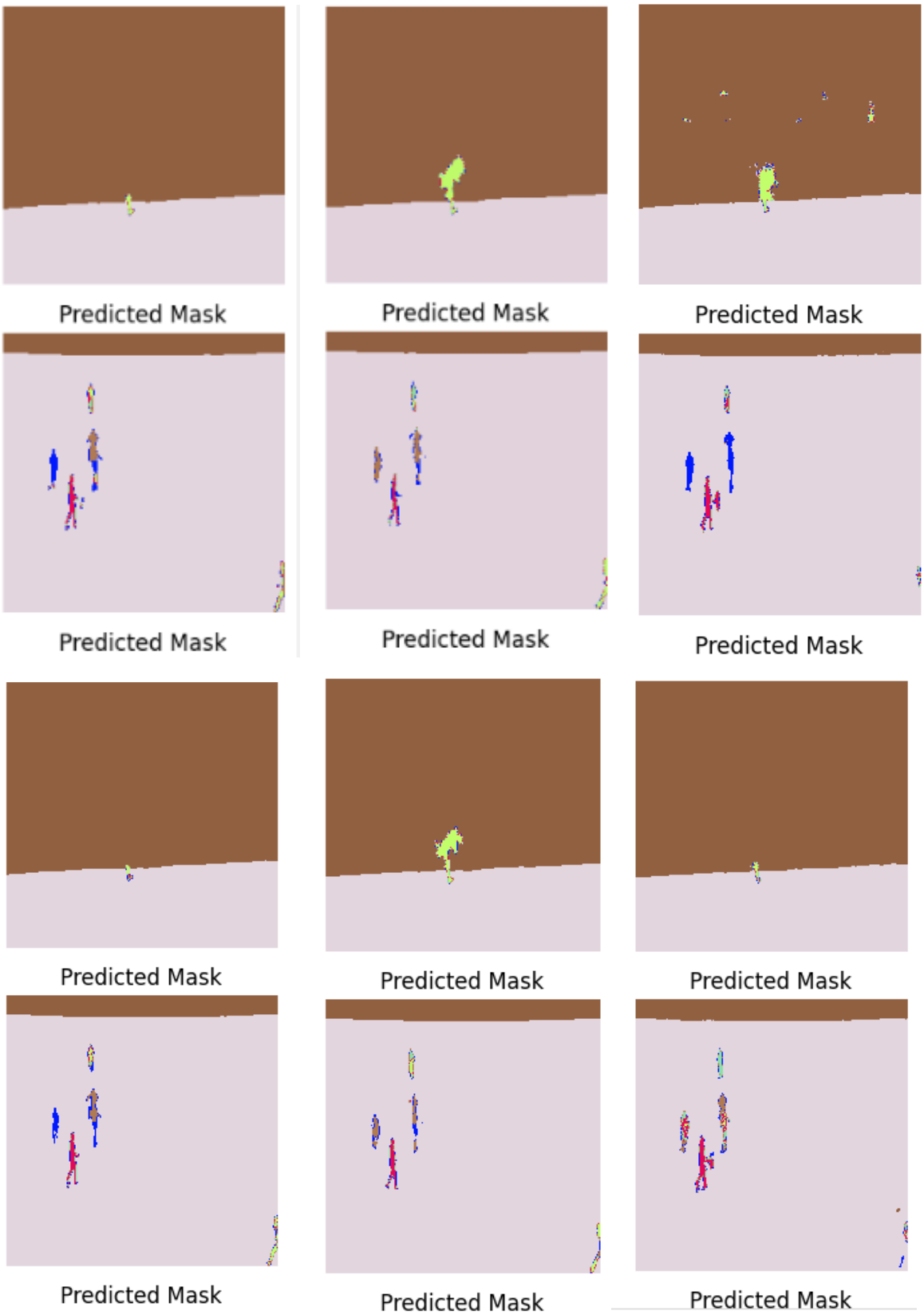


Ground Truth Mask

Original Image

Ground Truth Mask

Slika 11.


Predicted Mask
Predicted Mask
Predicted Mask
Predicted Mask
Predicted Mask
Predicted Mask
Predicted Mask
Predicted Mask
Predicted Mask
Predicted Mask
Predicted Mask
Predicted Mask

Slika 12.

Zaključak

Na osnovu dobijenih rezultata prikazanih u tabelama 1 i 2, moze se zakljuciti da su performanse DeepLabvsPlus mreze znatno bolje u odnosu na UNet i SegNet, prilikom treninga sa fiksnom i promenljiom stopom ucenja. Ocekivano je da ce DeepLabv3Plus dati najbolje rezultate,a SegNet, koji se odlikuje efikasnoscu ali ne i tacnoscu najgore. Na osnovu nekih prikazanih rezultata u vidu slika, moze se pretpostaviti da je teze izvrsiti segmentaciju objekta koji se nalaze na pozadini, nego pozadini u vidu trave, zbog same teksture i boje tipa pozadine. Opadanje learning rate-a ima uticaj na performanse UNet-a i SegNet-a, ali su i dalje vrednosti mIoU i f1 score-a niske. Visoka tacnost piksela, pracenja relativnim niskim srednjim IoU, govori o poteskoci uspostavljanja granica objekta od strane modela. Nebalansiranost klasa moze imati uticaj na IoU. mIoU od 67.66% se moze smatrati relativno dobrim rezultatom pri resavanju ovakvog problema. Relativno malu broj slika u dataset-u takodje moze predstavljati problem, pa se pored odradjene augmentacije preporucuje upotreba novih.

Literatura

[1] -Kaggle dataset
https://www.kaggle.com/datasets/sadhliroomyprime/cricket-semantic-segmentation
[2] - O.Ronneberger, P.Fischer, T.Box, “*U-Net: Convolutional Networks for Biomedical Image Segmentation*”
[3] - V.Badrinarayanan, A.Kendall,R.Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation”
[4] - L. Chen, G,Papandreou, F.Schroff, H.Adam, “Rethinking Atrous Convolution for Semantic Image Segmentation”