Univerzitet u Nišu, Elektronski fakultet
Katedra za računarstvo

Segmentacija puteva sa satelitskih snimaka
-Izveštaj projekta-

Predmet: Duboko Učenje

Profesor: prof. dr Aleksandar Milosavljević

Studneti: Predrag Nikolić 653
Sreten Šikuljak 1085

Niš, 2021

Sadržaj

1 O problemu

Ovaj projekat se bavi detekcijom puteva sa satelitskih snimaka. Problem je predsavljen u okviru *kaggle* takmičenja, koje može da se nađe na ovom linku. Takmičenje se bavi detekcijom puteva na satelitskim snimcima države Masačusec (savezna država u SAD-u). Dataset je napravljen za potrebe doktorskog rada[1] na temu segmentacije puteva.

Segmentacija puteva na osnovu satelitskih snimaka je izazovan zadatak. Postoji veliki broj izazova, tehničkih i u samim podacima. Problemi pri segmentaciji mogu biti obstrukcija od strane drveća, senke zgrada, različitost u teksturi i boji samih puteva. Takodje je bitan problem disbalans klasa, jer klasa puta predstavlja relativno mali broj pixela u odnosu na celu sliku.

2 Metode rešavanja problema

Doktorski rad[1] iz koga potiče dataset se bavi razvojem i evolucijom metoda koji mogu da se koriste za rešavanje problema kojim se bavi ovaj projekat. Ovaj problem može da se posmatra kao klasifikacija svakog piksela slike, gde bi se na izlazu svaki piksel klasifikovao kao put ili pozadina. Problem ovog pristupa je što bi ulazni vektor kao i izlaz iz modela bili preveliki čak i kada bi se smanjila slika, čime bi se izgubili podaci u procesu i segmentaciona mapa bi morala da se *resize*-uje na originalnu veličinu.

Napretkom hardvera je došlo do velikog napretka u istraživanju i tehnologijama vezanim za neuronske mreže, pa je to uslovilo nastanak modela koji kombinuju tehnike iz oblasti obrade slike, da bi radili nad slikama. Ovi modeli su bazirani na konvoluciji, a oblast koja se bavi obeležavanjem oblasti na slikama je semantička segmentacija.

2.1 Semantička segmentacija i Unet

Segmentacija je proces dodeljivanja svakog piksela slike nekoj klasi i na taj način se može posmatrati kao problem klasifikacije na nivou piksela. Postoje dva tipa segmentacije:

- Semantička segmentacija (*Semantic segmentation*) je proces klasifikovanja svakog piksela određenoj klasi. Svim pikselima iste klase se dodeljuje ista labela, pri čemu se ne vodi računa kojem objektu tačno taj piksel pripada. Na primer ako postoje dva čoveka na slici, svi pikseli koji pripadaju ljudima biće isto obeleženi.

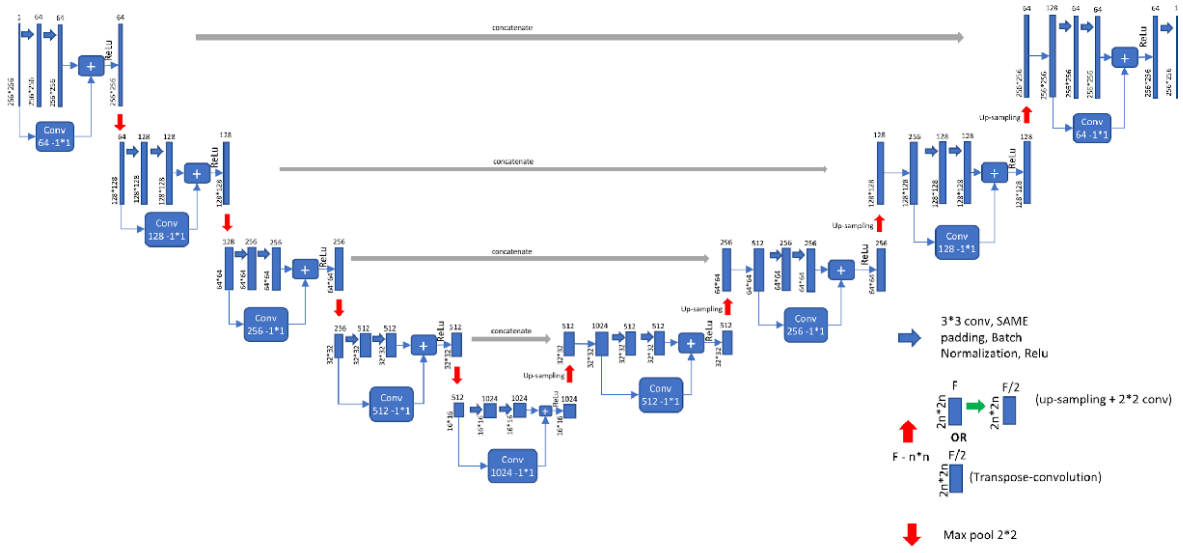
- Segmentacija instanci (*Instance segmentation*) se razlikuje od semantičke segmentacije po tome što se svakoj instanci objekta dodeljuje posebna labela. Ukoliko na slici nalaze dva čoveka, pikseli jednog čoveka će imati jednu labelu, a pikseli drugog čoveka drugu labelu.

Modeli koji se koriste za semantičku segmentaciju bazirani su na konvolucionim mrežama. Za segmentaciju može da se koristi model sa nekoliko konvolucionih slojeva, ali se koriste modifikacije ili unapređenja modela koji se koriste za klasifikaciju slika. Neki od modela koji se koriste za segmentaciju su:

- FastFCN – kombinacija *fully connected* slojeva i CNN arhitekture
- Gated-SCNN – Arhitektura sa dve odvojene CNN mreže čije se izlazi spajaju na kraju
- DeepLab – Bazirana na convolucionim slojevima

- Mask R-CNN – Nadogradnja Faster R-CNN modela
- Unet – Encoder/Decoder arhitektura

U okviru ovog projekta je korišćena Unet arhitektura, jer se ta arhitektura pokazala kao SOTA (*State-of-the-art*) rešenje u oblasti semantičke segmentacije. Unet pripada *encoder/decoder* arhitekturi, gde je *encoder* zadužen za izvlačenje karakteristika (*features*) sa slike, dok je *decoder* zadužen za rekonstrukciju segmentacione mape iz dobijenih karakteristika originalne slike. *Encoder* i *decoder* mogu biti sastavljeni od proizvoljnog broja blokova. Jedan blok enkodera se sastoji od dva konvoluciona sloja i jednog *max pool* sloja, dok se jedan blok dekodera sastoji od od dva konvoluciona i jednog sloja transponovane konvolucije. Broj filtara može da varira u kovolucionim slojevima, ali se obično prepolovljuje u encoderu, a duplira u dekoderu u susednim blokovima. Takođe odgovarajući blokovi enkodera i dekodera su povezani *skip* konekcijama, koje određene inforamacije koje daje enkoder prosleđuju dekoderu u fazi rekontrukcije slike.



Slika 1. Arhitektura proizvoljne Unet mreže

Sobzirom da je enkoder deo arhitekture, po strukturi isti kao arhitekture koje se koriste za klasifikaciju slika, on može da se zameni sa nekim modelom za klasifikaciju, tako što se modelu za klasifikaciju uklanjaju poslednji slojevi koji služe za pravljenje izlaza. Time se dobija model koji kao izlaza daje *feature* vektor, koji se povezuje u dekoder deo. Odgovarajući blokovi tog modela se povezuju sa odgovrajućim blokovima u dekoderu da bi se dobila Unet arhitektura.

Poenta korišćenja drugih modela kao enkodera u Unet-u je da se za enkoder iskoristi model koji je već istraniran na nekom većem skupu podataka (kao što je *imagenet*) , kako bi se na izlazu iz enkodera dobili bolji *feature* vektori. U tom slučaju mogu da se koriste težine modela bez modifikacije ili da se te težine koriste kao početna tačka u novom treniranju.

U ovom projektu je esperimentisano sa nekoliko modela različitih arhitektura za klasifikaciju kao što su *resnet*, *inception* i *vgg*.

2.2 Loss funkcija i metrika

Neke od loss funkcija koje se koriste u semantičkoj segmentaciji su:

- Softmax
- Jaccard
- Binarna (Kategorička) krosentropija
- Dice loss

Prilikom korišćenja modela za segmentaciju se uglavnom koriste binarna ili kategorička krosentropija, u zavisnosti od broja klasa. Međutim na ovom projektu je korišćen dice loss funkcija, zbog prirode skupa podataka. S obzirom da pikseli koji označavaju put čine znatno manji deo slike, dolazi do disbalansa kalsa, gde je pozadina (background) mnogo više zastupljena nego put. U tom slučaju bi rezultati bili loši, jer bi veći doprinos treniranju daala pozadina umesto ciljne klase puta.

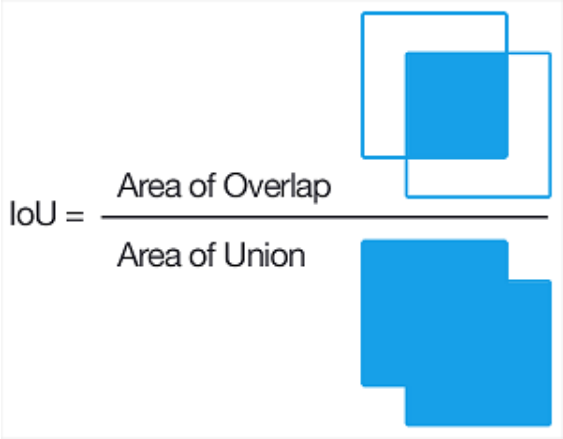
U ovom projektu je odabrano da se koristi dice loss, jer on meri presek predviđene i originlane klase za sve piskele i na taj način daje meru otpornu da disbalans klasa.

$$\frac{2 * |X \cap Y|}{|X| + |Y|}$$

Slika 2. Dice loss koeficijent

Konačna vrednost kao komplement u odnosu na 1 koeficijenta na slici 2, a takođe se obično dodaju parametri normalizacije na koeficijenat.

Standardna mera preciznosti koja meri procenat pogođenih piksela je ovde zamenjena merom IoU (*intersection over union*). Ideja je ista kao i za loss funkciju. Korišćenjem standarnde mere za preciznost rezultati nebi bili reprezentativni jer se klasa puta javlja mnogo manje u odnosu na pozadinu. Mera IoU se računa kao količnik preseka i unije, predviđene klase i originlane klase, za sve piksele tih klasa.



Slika 3. Intersection over union

2.3 Tehnologije

Projekat je implementiran u programskom jeziku *Python*, korišćenjem *Google open source* platforme *TensorFlow* i biblioteka *Keras, OpenCV* i drugih standardnig biblioteka koje se koriste u ML projektima. Takođe je iskorišćena javno dostupna biblioteka segmentation_models koja sadrži implementaciju Unet modela sa pretrniranim enkoderima. Projekat je implementiran kroz niz skpriti koje služe za treniranje i prirpemu dataseta, a pregled rezultata je dat u okviru *python notebook*-a.

Skripta *deleteWhiteImages.py* služi za brisanje nepotpunih slika koje imaju potpune maske. Parametri ove skripte su:

- putanja do folders sa slikama
- putanja do foldera sa odgovarajućim maskama
- putanja do tekstualnog fajla gde su yapisana imena slika koje treba da se obrišu

Svi ovi parametri se podešavaju u okviru fajla *config/deleteWhiteImages.yaml*.

Skripta *augmentDataset.py* služi za kreiranje dataseta za treniranje iz orignalnih slika. Parametri ove skripte su:

- putanja do originalnih slika
- putanja do originalnih maski
- putanja gde će biti smešten novi dataset
- procenti (0-100) za train, val i test skup
- broj *tile*-ova iz slike
- dimenzija *tile*-ova
- minimalna granica belog piksela za pretvranje u labelu puta(0-255)

Svi ovi parametri se podešavaju u okviru fajla *config/augment.yaml*.

Skripta *trainModel.py* služi za treniranje modela. Parametri ove skripte su:

- putanje do slika i odgovarajućih maski za train set
- putanje do slika i odgovarajućih maski za val set
- putanje do slika i odgovarajućih maski za test set
- putanja do modela ako se treniranje nastavlja
- parametri za unet
- dimenzije slike na ulazu
- learning rate
- broj epoha
- batch size
- da li se učitava prethodni model (true false)
- parametri za logovanje i čuvanje modela

Implementacija i rezultati su dostupni na javnom github repozitorijumu.

3 Skup podataka

Skup podataka sadrži 1171 sliku, dimenzija 1500x1500 piksela, koji pokrivaju prostor od 2.25 kilometra kvadratna. Skup je dostupan u okviru kaggle takmičenja. Za potrebe takmičenja skup podataka je podeljen na trening skup (1108 slika), validacioni skup (14 slika) i test skup (49 slika). Svaka slika sadrži odgovarajuću binarnu mapu na kojoj su obeleženi putevi belom bojom na crnoj pozadini.

Slika 4. Slika i odgovarajuća binarna mapa

Za potrebe projekta urađeno je određeno preprocesiranje podataka. Skup podataka sadrži određeni broj slika kojima fali deo informacija (deo slike koji je prikazan belom bojom), a imaju odgovarajuće mape na kojima su ucrtani putevi za delove koje ne postoje na slikama. Ove slike su uklonjene iz skupa podataka kako ne bi loše uticale na proces učenja modela. Spisak ovakvih slika se nalazi na repozitorijumu u okviru fajla *imgList.txt*, a slike mogu da se uklone pokretanjem skripte *deleteWhiteImages.py*. U ovom procesu uklonjeno je oko 300 slika.

Slika 5. Slika kojoj nedostaju podaci i mapa sa obeleženim putevima

Sobzirom da po incijalnoj podeli validacioni skup koji čini 2% od ukupnog skupa, da bi bio napravljen bolji balans, što kasnije utiče na proces treniranja i validnost rezultata, za potrebe ovog projekta sve slike se nasumično dele u *train/validation/test* skup sa odnosom 70%-20%-10%.

Pošto su dimenzije originalnih slika prevelike, model se trenira slikama dimenzije 256x256. Ove slike su dobijene sečenjem nasumično odabranog kvadrata dimenzije 256x256 iz originlanih slika. Na ovaj način su sačuvani originslni podaci is slika, a smnjena je dimenzija slike, bez transformacije piksela (*resize*). Da bi se pokrila cela originalna slika iz svake originalne slike se uzimaju 40 nasumično odabranih *tile*-ova. Nad svakim *tile*-om se sa određenom verovatnoćom vrši jedna od transforamcija:

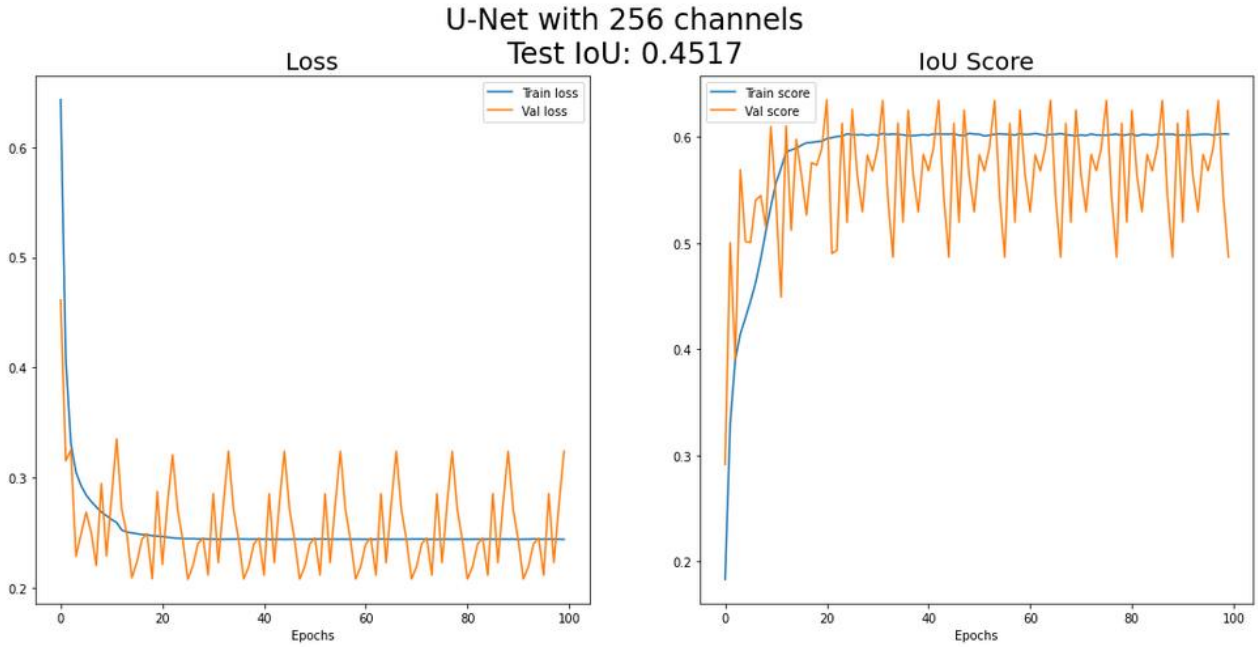
- Horizontalni flip
- Vertikalni flip
- Rotacija od 90, 180, 270 stepeni

Ove transformacije se dodaju kako bi uticale na raznovrsnost skupa i bolje obučila model za prepoznavanje. Augmentacija i kreiranje skupa podataka koji će da se koristi u procesu treniranja i

validacije se kreira od stratnog skupa pokretanje skripte *augmentDataset.py*. Nakong augmentacije se dobija skup podataka koji sadrži oko 25000 slika dimenzije 256x256 u trening skupu podataka.

4 Rezultati eksperimenata

Prvi korak u eksperimentima bio je uspostavljanje osnove. Napravljena je bazična U-Net arhitektura , bez pretreniranog encodera, sa početnom dubinom od 256. Podela skupa podataka za treniranje nije menjana od onog predstavljenom u takmičenju. Iz svake slike je nasumično izabrano 25 *tile*-ova, a jedan batch je sadržao 16 *tile*-ova, što je bio diktirano od strane hardverskih ograničenja.



Slika 6. Grafik loss-a i IoU mere na Unet256 modelu

Nakon inicijalnog eksperimeta isprobane su kombinacije veličine batch-a i kompleksnije mreze (UNet 512). IoU mera bila je nestabilna u toku treninga i validacije što je ukazivalo na raznovrstan ali nedovoljno obiman skup za validaciju. Samim tim mera koja se dobija na test setu ne oslikava pravu sposobnost mreže. Zbog toga je povećan broj *tile*-ova po slici i načinjena je predhodno pomenuta 70/20/10 podela na trening/validacioni/test skup.

Network	split	number_tiles	bs	lr	IoU	loss	IoU	loss	IoU	loss
UNet512	old	40	8	0.0001	0.6471	0.2169	0.8265	0.1391	0.6344	0.2622
UNet512	new	40	8	0.0001	0.7097	0.1757	0.6359	0.2365	0.5647	0.3312
UNet512	old	25	8	0.0001	0.6471	0.2169	0.8265	0.1391	0.5321	0.3613
UNet256	old	25	16	0.0001	0.6019	0.2439	0.626	0.2077	0.4517	0.4416
UNet256	new	40	16	0.0001	0.6616	0.2109	0.6354	0.2436	0.5924	0.3185

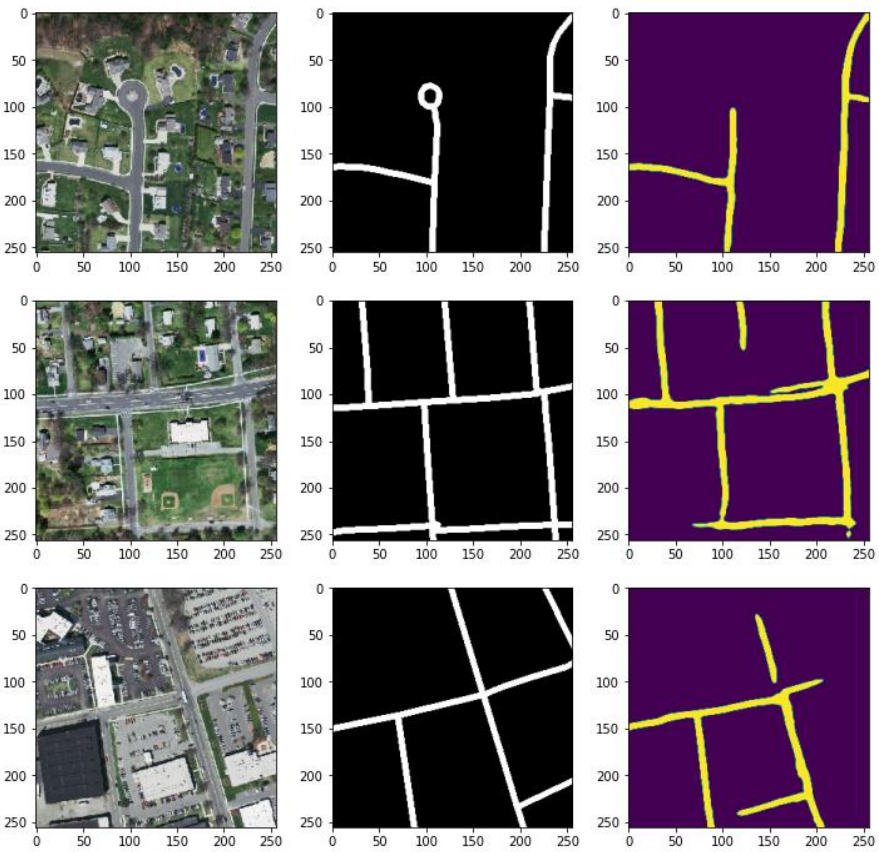
Tabela 1. U-Net bez pretreniranog encodera

Naredni eksperimenti su sprovođeni sa encoderima pretreniranim na *imagenet* skupu podataka. Kao što je i čekivano, rezultati na test skupu su bolji a trening brže konvergira. Od parametara treninga eksperimentisano je sa veličinom jednog *batch*-a, koeficijentom učenja i slobodom trenriranja pretreniranog encodera.

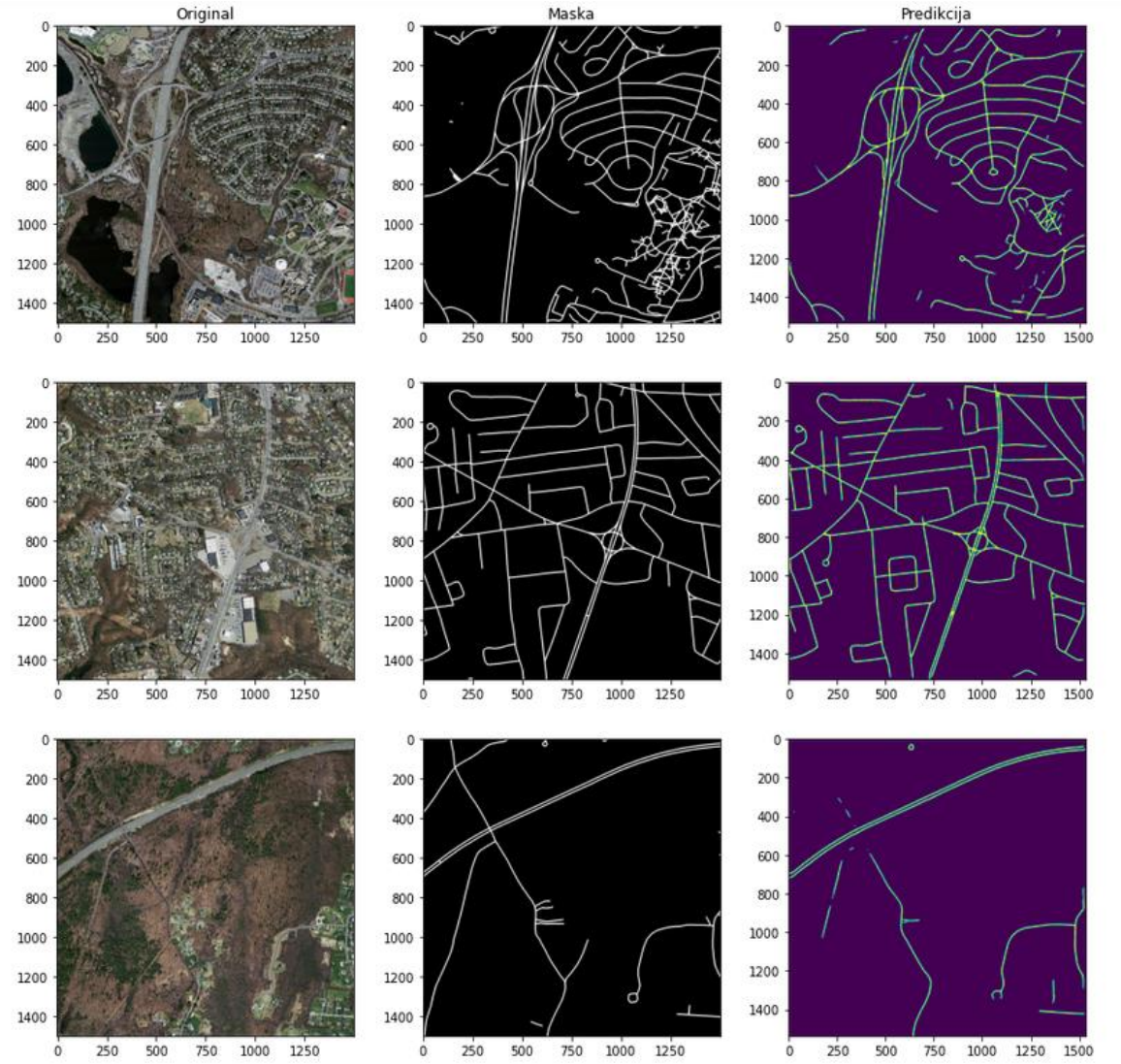
Hyper-parameters					Training		Val		Test	
Encoder	Decoder	Trainable	bs	lr	IoU	loss	IoU	loss	IoU	loss
Inceptionv3	256-16	TRUE	16	0.0001	0.8915	0.1005	0.6156	0.2369	0.6117	0.2425
	512-16				0.8198	0.1001	0.6179	0.2378	0.6133	0.2411
Inceptionresnetv2	256-16	TRUE	16	0.0001	0.8228	0.0987	0.6195	0.2365	0.6155	0.2397
vgg16	256-16	TRUE	16	0.0001	0.7934	0.1163	0.619	0.2371	0.616	0.2393
			32		0.8052	0.109	0.6112	0.2423	0.6102	0.2431
resnet18	256-16	TRUE	16	0.0001	0.7958	0.1151	0.6047	0.2479	0.6025	0.2498
		FALSE			0.7155	0.1671	0.5642	0.2804	0.5594	0.2845
resnet34	256-16	FALSE	16	0.0001	0.7122	0.1694	0.5697	0.2758	0.5658	0.2794
resnet50	256-16	FALSE	16	0.0001	0.7371	0.1526	0.5812	0.2665	0.5771	0.2702
		TRUE	8	0.0001	0.7379	0.1527	0.6197	0.2379	0.6167	0.2386
			16	0.0001	0.8153	0.1028	0.61683	0.2383	0.6152	0.2396
				8.00E-05	0.7619	0.1361	0.618	0.2376	0.6185	0.2374

Tabela 2. Trening U-Net arhitekture sa pretreniranim encoderom

Najveći uticaj na rezultat je imala sloboda treniranja encodera, dok veličina *batch*-a i keoficijent učenja nisu davali značajne promene. Oni modeli koji nisu mogli da adaptiraju encoder su imali ~5-7% losiju IoU meru. To je posledica činjenice da satelitske slike nisu deo *imagenet* skupa podataka, pa je potrebno da encoder nauči kako da kombinuje karakteristike niskog nivao na relevantan način za ovaj problem.

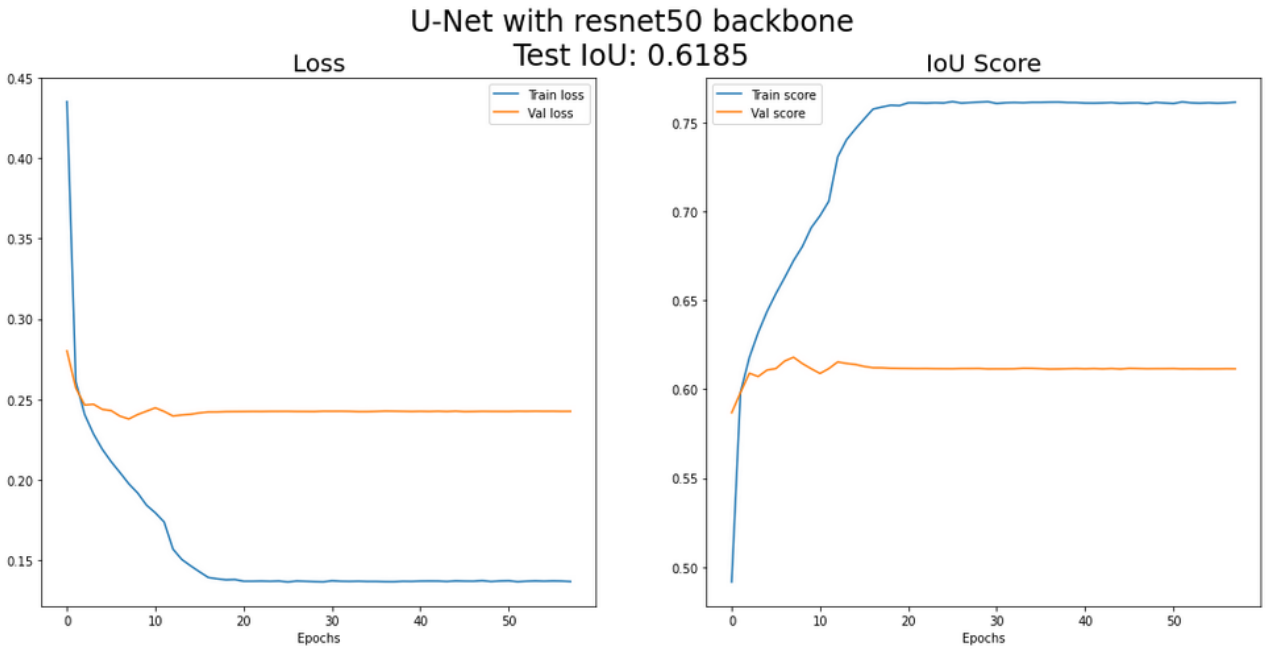


Slika 7. Segmentacija nad 256x256 slikama



Slika 8. Segmentacija nad originalnim slikama 1500x1500

Model sa najboljom IoU merom je treniran sa odmrznutim ResNet50 encoderom i veličinom *batch*-a 8. Kao sto se na Slici 8. vidi, segmentacija na nivou jedne 256x256 slike potpuno oslikava IoU meru koja je dobijena na testu. Senke, parkinzi, drveće i makadamski putevi predstavljaju poteskoće modelu kako da ih segmentira, pre svega zbog loših segmentacionih maski na nivou skupa podataka. Kada se segmentacija vrši na nivou čitave slike, rezultati postaju bolji. Greške jos uvek postoje, međutim, na nekim primerima se postavlja pitanje čija je greska bila, anotacije ili modela, što se može videti iz slike 8.



Slika 9. Grafik lossa i IoU mere za najbolji model

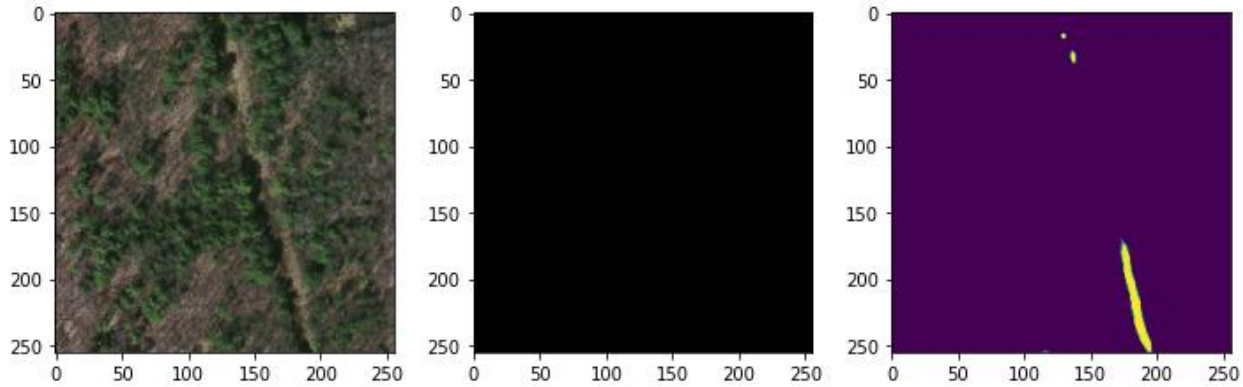
5 Zaključak

Kao što je prethodno prikazano u okviru ovog projekta izvršeno je više eksperimenata sa različitim modelima baziranim na Unet arhitekturi. Eksperimentisano je sa brojem slojeva i brojem filtara u konvolucionim slojevima Unet arhitekture. Takođe eksperimentisano je raznim mrežema koje se koriste kao enkoderi u Unet arhitekturi, kao što su *resnet*, *inception* i *vgg*. Tako đe eksperimentisano je sa parametrima treninga (*batch size, learning rate*) kao i sa zamrzavanjem i naknadnim treniranjem težina pretreniranih modela koji su se koristili kao enkoderi.

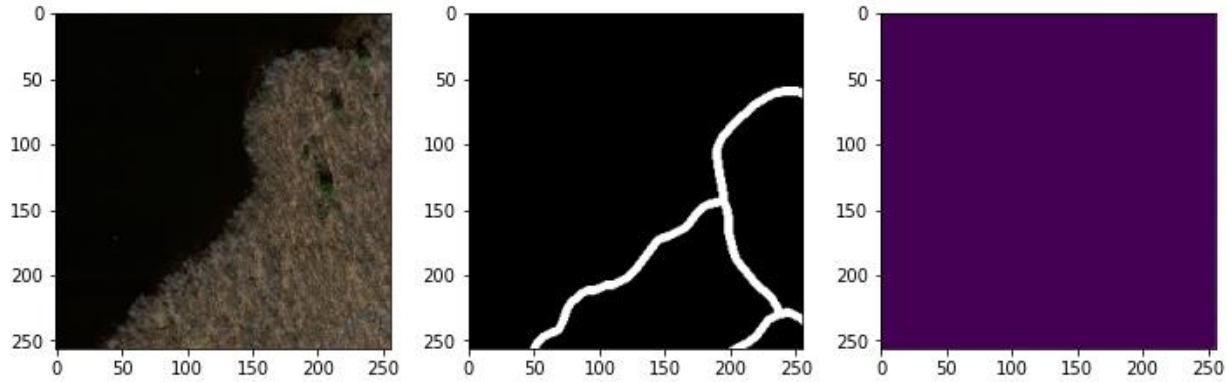
Kao što možemo da vidimo eksperimenti ne variraju mnogo u rezultatima, najbolji modeli postižu oko 61% na validacionom skupu i takođe oko 61% na test skupu. Korišćena mera je IoU (*intersection over union*). Postignuti rezultati svakako nisu sjajni, ali postoji prostor za poboljšanje. Takođe bitno je napomenuti da su rezultati postignuti na modelima treniranim na slikama veličine 256x256, tako da neki sitni nedostaci nisu vidljivi na originalnim slikama dimenzije 1500x1500.

Jedan od glavnih problema korišćenog skupa podataka su loše obeležene maske. Ove maske su nastale pristupanjem nekog API-ja za javno dostupne maps, koje nemaju sve informacije o ulicama. Naime u datasetu postoje slike na kojima nisu obeležene ulice ili delovi ulica koji nisu obeleženi, postoje zemljani putevi koji su na nekim mestima obeleženi, a na nekim ne i postoje putevi koji su kompletno prekriveni drugim objektima. Sve ovo utiče na to da model ne može da konvergira u procesu treniranja ka datom cilju, što rezultuje ovakvim rezultatima na validacionom i trening skupu. Takođe možemo da primetimo da rezolucija ako velika za originalne slike nike dovoljna da bi se neki putevi jasno videli.

Slika 10. Cela ulica nije obeležena na labeli



Slika 11. Zemljani put koji nije obeležen na labeli. Model pokušava da ga obeleži kao put



Slika 12. Put je obeležen na labeli ali je jedva vidljiv na slici. Model vidi sve kao pozadinu (background)

Jedan od načina za poboljšanje rezultata bi bio ponovno labelovanje skupa podataka, gde bi se na postojećim labelama ispravili pomenuti nedostaci. Time bi se uvela stablinost u proces treniranja i loss parametar bi mogao još više da se smanji.

Takođe neki od generalnih problema detekcije puta se ogeldaju u tome što put može delimično da bude prekriven objektima (šuma, automobili), takođe parkinzi, piste na aerodromu, prilazi do garaža su prikazani istim piskelima kao i put, što pravi problem modelu u procesu treniranja.

Što se tiče eksperimenata možemo da primetimo da kada se koriste pretrnirani modeli kao enkoderi u Unet arhitekturi, u procesu treniranja dobijamo znatno veći skor na treningu u odnosu na validacioni skor, gde bi smo mogli da kažemo da model *overfitt*-uje. Razlog je to što su ti modeli trenirani na mnogo većem i raznovrsnije skupu slika, pa prilikom izvlačenja *high level feture*-a, slike iz korišćenog skupa podataka mogu da izgledaju slično.

Ovaj problem bi mogao da se prevaziđe tako što bi se iz istreniranih modela korsitilo prvih nekoliko slojeva, umesto cele mreže, na koje bi se povezalo još nekoliko blokova Unet-a i tako napravio enkoder deo. Problem je odrediti koliko slojeva trenirane mreže je optimalno, pa bi ova teorija zahtevala još dosta eksperimenata, kako bi se utvrdila njena tačnost.

Model bi takođe mogao da se poboljša i generalizuje dodavanjem slika iz drugih datasetova koji sadrže satelitske snimke, koji sadrže odgovarajuće mape na kojima su obeleženi putevi. Postoji nekolicina takvih skupova podataka koji su javno dostupni, a sadrže segmentacione maps na kojima su obeleženi putevi ili više klasa.

Literatura

[1] V. Mnih, "Machine Learning for Aerial Image Labeling", University of Toronto, 2013
[2] Derrick Mwiti, Katherine (Yi) Li, "Image Segmentation in 2021: Architectures, Losses, Datasets, and Frameworks", 2021
[3] O. Ronneberger, P. Fischer, T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", 2015
[4] Jerin Paul, "Segmentation of Roads in Aerial Images.", 2019
[5] Literatura sa kursa "Duboko učenje" – Elektronski Fakultet u Nišu, 2021