

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1451

**ANALIZA SATELITSKIH SNIMAKA ZA KLASIFIKACIJU
ŠUMSKIH VRSTA**

Matija Jakovac

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1451

**ANALIZA SATELITSKIH SNIMAKA ZA KLASIFIKACIJU
ŠUMSKIH VRSTA**

Matija Jakovac

Zagreb, lipanj 2024.

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1451

Pristupnik: **Matija Jakovac (0036538710)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentor: prof. dr. sc. Marko Subašić

Zadatak: **Analiza satelitskih snimaka za klasifikaciju šumskih vrsta**

Opis zadatka:

Ispitati metode za segmentaciju slika s naglaskom na segmentaciju slika vegetacije. Pripremiti odgovarajuću bazu slika iz višekanalnih satelitskih snimaka šumskog područja. Implementirati metodu za segmentaciju višekanalnih slika temeljenu na dubokim neuronskim mrežama. Provesti analizu korisnosti multi temporalnih višekanalnih satelitskih snimaka te odgovarajuću selekciju podskupa ulaznih slika. Provesti analizu točnosti segmentacije na prikupljenoj bazi slika. Ispitati ovisnost točnosti o razini detalja u satelitskim snimcima. Za implementaciju koristiti programski jezik Python.

Rok za predaju rada: 14. lipnja 2024.

Hvala Bogu, obitelji, prijateljima i mentoru na podršci

Sadržaj

1. Uvod	3
1.1. Pozadina i motivacija	3
1.2. Ciljevi rada	3
1.3. Struktura rada	4
2. Pregled literature	5
2.1. Analiza postojećih modela za segmentaciju	5
2.1.1. Segmentacija	5
2.1.2. Najčešće korišteni modeli	6
2.2. Korištenje Sentinel satelita	6
3. Umjetne neuronske mreže	7
3.1. Općenito	7
3.2. Učenje neuronske mreže	7
3.3. Umjetni neuron	9
3.4. Implementacija	11
3.5. Konvolucijske neuronske mreže	11
3.5.1. Konvolucija	11
3.5.2. Arhitektura mreže	11
4. Skup podataka	14
4.1. Satelitske snimke	14
4.2. Segmentacijske maske	15
5. Metodologija	17
5.1. Korištene tehnologije	17

5.2.	Pretprocesiranje podataka	17
5.3.	UNET model	21
5.3.1.	Općenito	21
5.3.2.	Arhitektura UNET modela	22
5.4.	Trening i evaluacija	24
5.4.1.	Hiperparametri modela	24
5.4.2.	Funkcija greške i optimizator	26
5.4.3.	Dodatna poboljšanja	27
5.4.4.	Evaluacijske metrike	28
6.	Rezultati	31
6.1.	Prikaz i rasprava o rezultatima modela	31
7.	Zaključak	39
	Literatura	41
	Sažetak	44
	Abstract	45

1. Uvod

1.1. Pozadina i motivacija

Jedna od ključnih metoda za praćenje i klasifikaciju šumskih vrsta na Zemljinoj površini analiza je satelitskih snimaka. Takve se snimke koriste upravo zbog visoke sposobnosti satelita da pokrije iznimno velike geografske površine. Snimke koje nam satelit omogućava vrlo su visoke rezolucije što olakšava i poboljšava njihovu analizu. Metoda analize omogućava ekolozima, znanstvenicima i istraživačima da u stvarnom vremenu prate promjene u šumskim ekosustavima kao što su degradacija okoliša i deforestacija. Značaji kojim ovakva analiza pridonosi su ekološki te ekonomski.

S pogleda ekološkog značaja pomaže se u zaštiti bioraznolikosti, očuvanju prirodnog staništa i upravljanju resursima u održavanju šuma. Ekonomski, praćenjem šumskih vrsta pomaže se u planiranju gospodarskih aktivnosti poput šumarstva i poljoprivrede. Troškovi povezani uz prirodne katastrofe kao što su požar i poplava također se mogu smanjiti ispravnom analizom snimaka.

1.2. Ciljevi rada

Cilj je ovog rada razviti i ispitati UNET model za segmentaciju i klasifikaciju šumskih vrsta korištenjem *Sentinel* satelitskih snimaka. Primjenjivanjem raznih hiperparametara modela cilj je poboljšati njegovu točnost i učinkovitost prilikom segmentacije. Ispitivanje odnosno evaluacija modela uključivat će usporedbu dobivenih rezultata sa stvarnim rezultatima te analizu prednosti i nedostataka korištenja modela za ovaj problem. U suštini, cilj je razviti model koji će u budućnosti biti koristan i primjenjiv za razne potrebe pritom pružajući točne i pouzdanje podatke kako bi se donijeli ispravni zaključci.

1.3. Struktura rada

Ovaj je rad podijeljen u šest poglavlja kako bi čitatelj postupno mogao razumjeti cilj i svrhu obrađivanja teme ovakve vrste. Nakon uvoda slijedi drugo poglavlje koje opisuje postojeće modele i njihovu primjenu na zadatke segmentacije kao i sam opis što je segmentacija. Poglavlje završava pregledom podataka prikupljenih od *Sentinel* satelita čiji su detalji i primjena također opisani. Tehnika pomoću koje će zadatak biti riješen opisana je u trećem poglavlju. Podatci koji će biti obrađeni analizirani su u četvrtom poglavlju. Nastavno, peto poglavlje opisuje razvoj i evaluaciju UNET modela kao i pretprocesuiranje slike. Rezultati modela opisani su u šestom poglavlju koje uključuje njihovu analizu i vizualizaciju. Sedmo poglavlje, koje je ujedno i zadnje poglavlje, ukratko prikazuje ključne rezultate te analizira rad u cijelosti ističući njegove prednosti i nedostatke.

2. Pregled literature

2.1. Analiza postojećih modela za segmentaciju

2.1.1. Segmentacija

Zadatak segmentacije u kontekstu analize slike predstavlja razdvajanje originalne slike u različite segmente odnosno dijelove koji svaki za sebe sadržava informaciju o određenom području slike. Jedna je od najčešće korištenih vrsti semantička segmentacija čiji model za ulaznu sliku stvara segmentacijsku masku (Slika 2.1.). Maska je zapravo rekonstrukcija originalne slike čiji je svaki piksel obojen u boju koja predstavlja pojedinačnu klasu. Modeli koji se upotrebljavaju za zadatak koriste neuronske mreže koje su detaljnije opisane u trećem poglavlju, a najčešće se upotrebljavaju u biomedicini, autonomnim vozilima, robotici te poljoprivredi.



Slika 2.1. Primjer semantičke segmentacije [1]

2.1.2. Najčešće korišteni modeli

Razvojem tehnologije dubokog učenja i neuronskih mreža nastali su mnogi modeli koji obrađuju zadatak segmentacije. Neki su od najpoznatijih potpuno povezane konvolucijske mreže (FCN) te SegNet, UNET i DeepLab modeli [1] [2] koji za svoju implementaciju koriste tradicionalne konvolucijske neuronske mreže. UNET se model razvio u jedan od najuspješnijih modela prilikom segmentacije slika u biomedicini. Njegova je vrlina također široka primjena i na ostala područja uključujući analizu satelitskih snimaka. Njegova arhitektura omogućava preciznu i ispravnu segmentaciju čak i kada su dostupni relativno mali skupovi podataka za trening.

2.2. Korištenje Sentinel satelita

Program Sentinel [3], kojeg provodi Europska svemirska agencija (ESA), sastoji se od mnogobrojnih satelita čija je uloga pružanje visokokvalitetnih podataka za potrebe proučavanja Zemlje (Slika 2.2.). Za proučavanje Zemljine površine najkorisniji su sateliti *Sentinel-1*, *Sentinel-2* te *Sentinel-3*. *Sentinel-2* služi za prikupljanje podataka neovisno o vremenskim uvjetima dok ostala dva koriste visokokvalitetne senzore kako bi prikupili visoko rezolucijske optičke podatci. Takvi su podatci od ključne važnosti za praćenje vegetacije, u ovom slučaju za praćenje šumskih vrsta.

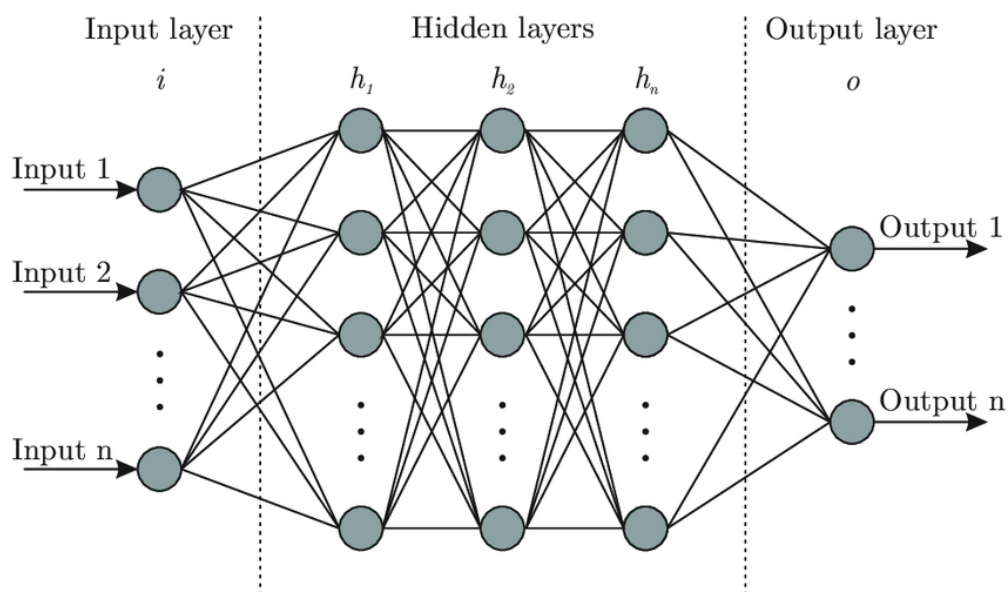


Slika 2.2. Program Sentinel [3]

3. Umjetne neuronske mreže

3.1. Općenito

Umjetna neuronska mreža [4] [5] skup je međusobno povezanih umjetnih neurona čija se funkcionalnost temelji na biološkom neuronu odnosno uređena je po uzoru na ljudski mozak (Slika 3.1.). Njihovom interakcijom moguće je obraditi veliki skup podataka te su sposobni učiti iz njih i donositi kvalitetne zaključke.



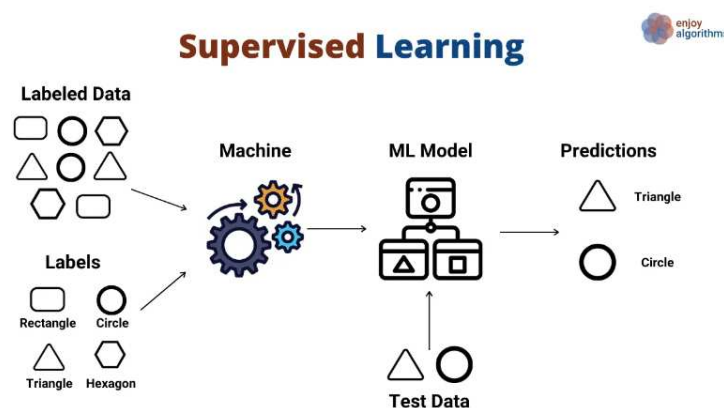
Slika 3.1. Umjetna neuronska mreža [6]

3.2. Učenje neuronske mreže

Učenje neuronske mreže dijeli se na etapu učenja odnosno treniranja te etapu obrade podataka odnosno iskorištavanja naučene mreže. Predočavanjem ulaznih primjera i njihovog očekivanog izlaza može se postupno prilagoditi težina veze između neurona. Takvo jedno predočavanje svih uzoraka u skupu podataka naziva se epohom. Učenje se

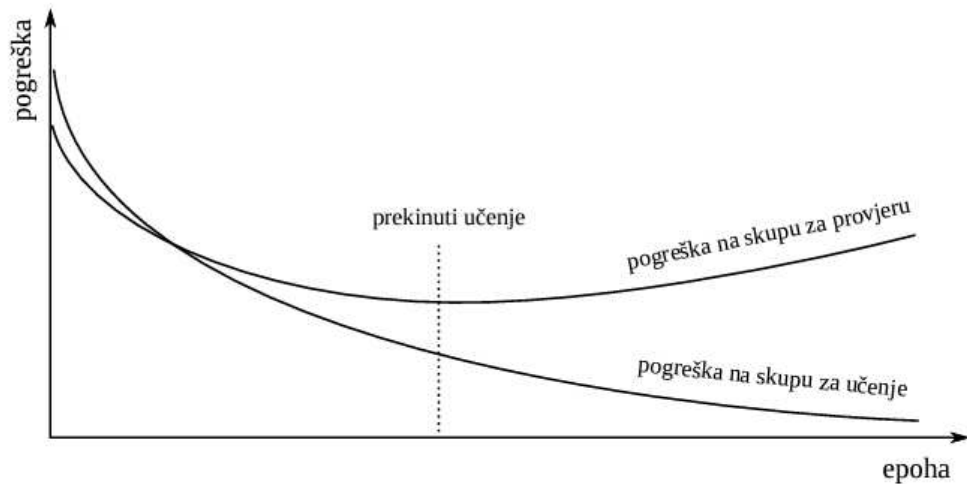
dijeli na pojedinačno učenje, učenje s minigrupama (eng. *mini-batches* i grupno učenje (eng. *batch*). U ovom će se radu koristiti učenje s minigrupama u kojem se učenje događa s više predočenih uzoraka radi ubrzanja procesa.

Postoje tri vrste učenja neuronskih mreža: nadzirano, nenadzirano i podržano. Nadzirano učenje [7] vrsta je učenja koja se zasniva na tome da se mreži preda predodređeni izlaz za svaki pojedinačni ulaz (Slika 3.2.). Pomoću toga znanja neuronska će mreža kroz iteracije mijenjati iznos svojih težina i pomaka.



Slika 3.2. Nadzirano učenje [7]

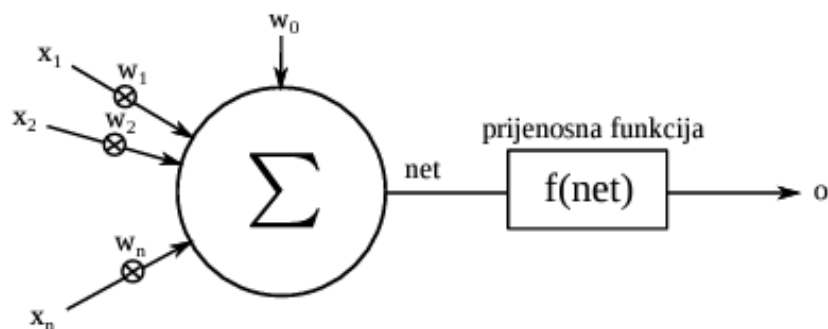
Od početnog skupa podataka za potrebe učenja izrađuju se nova tri skupa podataka. Najveći postotak, koji iznosi 70%, odnosi se na skup za učenje (eng. *training set*) u kojem se tijekom iteracija težine namještaju tehnikom vlastitog odabira kao što su propagacija pogreške unatrag (eng. *Error Backpropagation*) ili pomoću genetskog algoritma. Sljedećih 20% originalnog skupa podataka pripada skupu za provjeru (eng. *validation set*) koji se koristi uz skup za treniranje kako bi se provjerilo svojstvo generalizacije mreže. Koristeći taj skup podataka osigurava se da mreža ne poprimi svojstvo pretreniranosti. S tim svojstvom mreža gubi svojstvo generalizacije i odlično predviđa samo na podacima skupa za treniranje odnosno postaje štreber. U trenutku kada pogreška na skupu za provjeru počne rasti prekida se učenje (Slika 3.3.) te je mreža spremna za testiranje. Testira se na preostalih 10% uzoraka koje mreža nikad nije vidjela. Rezultati na tom skupu koriste se kako bi se mreža usporedila s mrežama drugačije arhitekture.



Slika 3.3. Graf pogreške na skupu za treniranje i provjeru [4]

3.3. Umjetni neuron

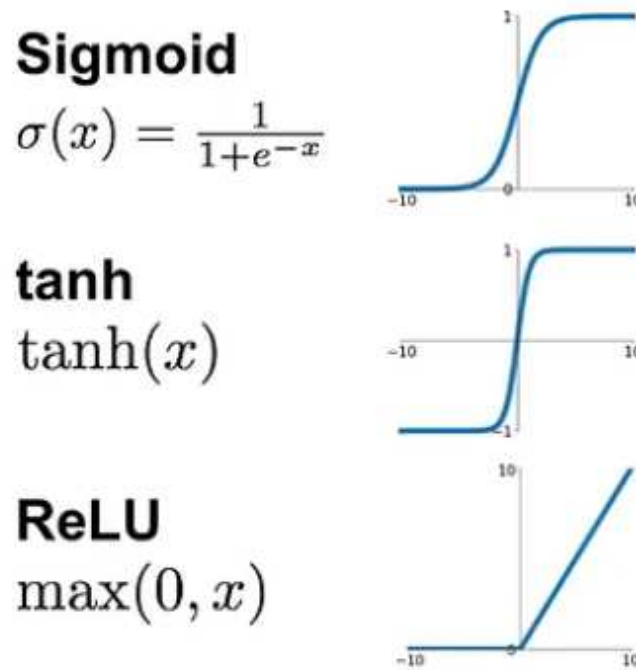
Umjetnom neuronu moguće je na ulaz postaviti jedan ili više signala koji se obrađuju pomoću težina i pomaka (eng. *bias*) te se aktivacijskom funkcijom šalju kao izlazni signal drugim neuronima. Vrijednost svakoga ulaza x_i množi se s težinom toga ulaza w_i te se na tu ukupnu sumu nadodaje pomak w_0 (Jednadžba 3.1). Ovakvim računanjem dobiva se ukupna vrijednost koja se označava s *net* (Slika 3.4.).



Slika 3.4. Umjetni neuron [4]

$$\text{net} = \left(\sum_{i=1}^n x_i \cdot w_i \right) + w_0 \quad (3.1)$$

Izlazna vrijednost neurona nastaje propuštanjem prijašnje vrijednosti kroz prijenosnu funkciju, $f(net)$, od kojih se izdvajaju sigmoidalna funkcija, tangens hiperbolni i zglobnica (eng. *Rectified Linear Unit*, ReLU). Grafovi navedenih aktivacijskih funkcija prikazani se na slici 3.5.



Slika 3.5. Aktivacijske funkcije [8]

Tangens hiperbolni funkcija definira se kao:

$$f(net) = \tanh(net) = \frac{2}{1 + e^{-2 \cdot net}} - 1 = 2 \cdot \sigma(2 * net) - 1 \quad (3.2)$$

Sigmoidna funkcija definira se kao:

$$f(net) = \sigma(net) = \frac{1}{1 + e^{-net}} \quad (3.3)$$

Zglobnica definira se kao:

$$f(net) = \text{ReLU}(net) = \max(0, net) \quad (3.4)$$

3.4. Implementacija

Neuronske mreže imaju različite vrste implementacije od čega se izdvajaju jednostavni perceptroni (Slika 3.1), višeslojni perceptroni (eng. *Multilayer perceptron*, MLP), rekurentne neuronske mreže (eng. *Recurrent neural network*, RNN) i konvolucijske neuronske mreže (eng. *Convolutional neural network*, CNN). Upravo je UNET model, koji se koristi u rješavanju zadatka, konvolucijska neuronska mreža čija se arhitektura opisuje detaljnije u sljedećem poglavlju. Svaka od navedenih arhitektura prilagođena je za specifični zadatak, poput klasifikacije, detekcije objekata, predikcije sekvenci te na posljetku segmentacije. Danas neuronske mreže pronalaze široku primjenu u raznim područjima, uključujući računalni vid i prirodni jezik, radi svoje sposobnosti za automatsko učenje iz velikih skupova podataka bez potrebe za programiranjem pravila od strane čovjeka.

3.5. Konvolucijske neuronske mreže

3.5.1. Konvolucija

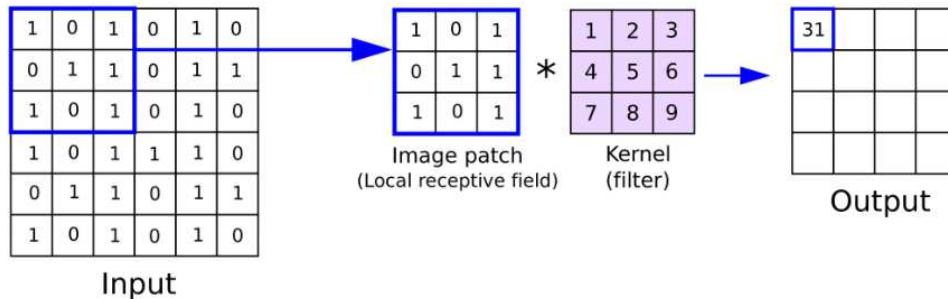
Konvolucija [1] [9] matematička je operacija dviju funkcija koja producira novu, treću, funkciju. Definirana je kao integral produkta dvaju funkcija nakon što se jedna od njih rotira oko y-osi te se translacija. Integral se računa za sve vrijednosti translaticirane funkcije čime nastaje prije spomenuta nova funkcija (Jednadžba 3.5). Odabir koja se od dvije funkcije ulaza rotira i translacija ne utječe na rezultat odnosno konvolucija ima svojstvo komutativnosti.

$$(f * g)(t) := \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau \quad (3.5)$$

3.5.2. Arhitektura mreže

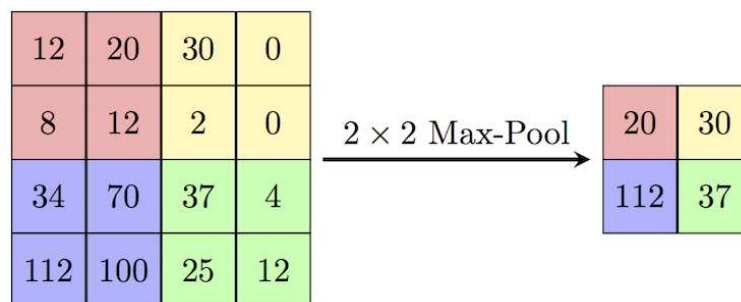
Konvolucijske neuronske mreže [1] [10] [11] tip su neuronske mreže koja uči značajke podataka pomoću takozvanog filtera (eng. *kernel*). U kontekstu slike jednu funkciju predstavljaju vrijednosti piksela na slici koju prekriva filter, a drugu upravo vrijednosti toga filtera (Slika 3.6.). Postupnim pomicanjem filtera po cijeloj slici i primjenjivanjem konvolucije nastaju mape značajki (eng. *feature maps*). One sadrže informacije o

rubovima i tekstura na slici. Postupak se može ponoviti nekoliko puta s novom veličinom filtera te time nastaju nove mape. Spajanjem svih mapa u jedan vektor nastaje izlaz prvog sloja mreže, konvolucijskog sloja.



Slika 3.6. Konvolucijski sloj [11]

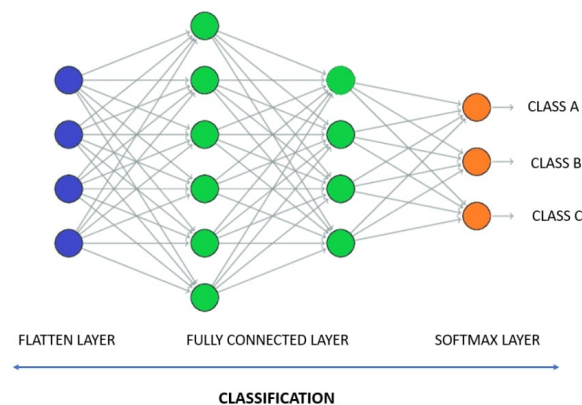
Često se između dva konvolucijska sloja nalazi sloj maksimalnog sažimanja (eng. *max-pooling layer*). Njegov su ulaz mape značajki iz prijašnjeg sloja te se na svakoj primjenjuje operaciju sažimanja. Tim postupkom smanjuje se rezolucija slika uz očuvanje najbitnijih karakteristika slike. Najčešće se slika podijeli u 2 x 2 matrice koje se ne preklapaju te se od te četiri vrijednosti izabere ona najveća (Slika 3.7.) koja se prosljeđuje i čini sliku smanjene rezolucije. Svrha je ovog sloja smanjenje parametara i nepotrebnog računa čime se poboljšava efikasnost mreže i izbjegava prenaučenosť.



Slika 3.7. Maksimalno sažimanje [1]

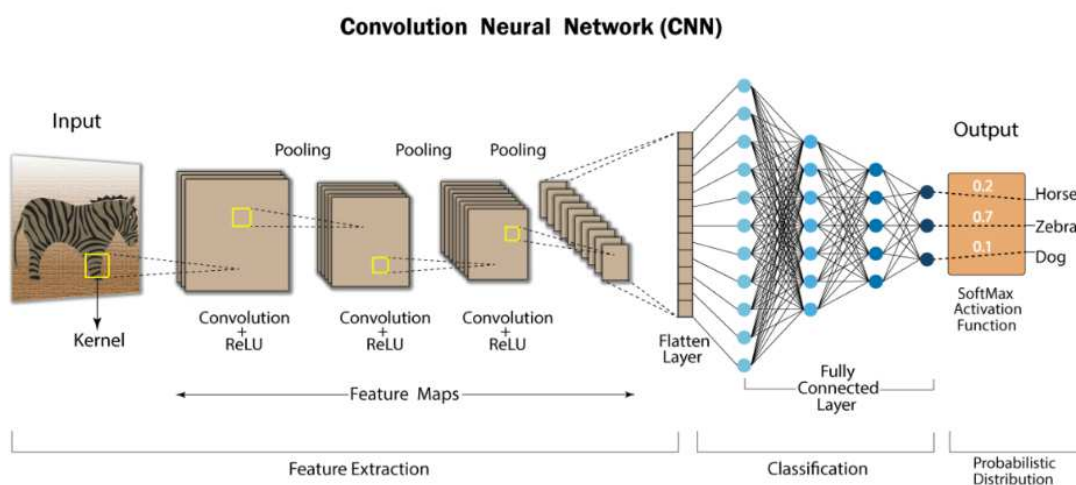
Sljedeći je sloj aktivacijski u kojem se primjenjuje aktivacijska funkcija koja je nelinearna te se zbog toga u mrežu uvodi nelinearnost. Također se omogućuje mreži da radi složenije poveznice između podataka na slici. Najčešće je korištena aktivacijska funkcija zglobnica (Jednadžba 3.4). Njome se sve negativne vrijednosti koje su dobivene na ulazu pretvaraju u nulu.

Zadnji sloj u mreži potpuno je povezani sloj (eng. *fully-connected layer*) s kojim završava svaka neuronska mreža, neovisno je li ona konvolucijska ili ne. Primjenom linearne kombinacije i po želji aktivacijske funkcije sloj od ulaznog vektora producira izlazni (Slika 3.8.). U zadatku segmentacije izlazni je vektor veličine N, gdje N označava broj klasa u zadatku. Svaki element vektora je vjerojatnost ulazne slike da pripada jednoj od zadanih klasa. Izračun se svodi na primjenu formule prije spomenutog jednostavnog perceptrona. Ako su točno zadane dvije klase koristi se logistička, a ako su više od dvije klase *softmax* aktivacijska funkcija.



Slika 3.8. Potpuno povezani sloj [12]

Spajanjem navedenih slojeva nastaje potpuno funkcionalna konvolucijska neuronska mreža (Slika 3.9.). Mreža se trenira koristeći tehniku propagacije unatrag pomoću koje se težine između neurona postupno izmjenjuju pronalazeći optimalno rješenje.



Slika 3.9. Konvolucijska neuronska mreža [13]

4. Skup podataka

4.1. Satelitske snimke

Uporabom *Sentinel-2* satelita u razdoblju od deset mjeseci tijekom 2020. godine snimljeno je deset snimaka. Svaka snimka sadrži dvanaest spektralnih pojasa (Tablica 4.1.) koji omogućavaju identifikaciju i mogućnost razlikovanja različitih vrsta vegetacije pomoću njihovih karakteristika.

Broj	S-2 Sloj	Opis	Srednja valna duljina (μm)
1	B01	Coastal aerosol	0.443
2	B02	Blue	0.49
3	B03	Green	0.56
4	B04	Red	0.665
5	B05	Red edge 1	0.705
6	B06	Red edge 2	0.74
7	B07	Red edge 3	0.783
8	B8A	Near infrared (NIR) 1	0.842
9	B09	Red edge 4	0.865
10	B10	Water vapour	0.945
11	B11	Short-wave infrared (SWIR) 1	1.61
12	B12	SWIR 2	2.19

Tablica 4.1. Spektralni pojasi Sentinel-2 satelita

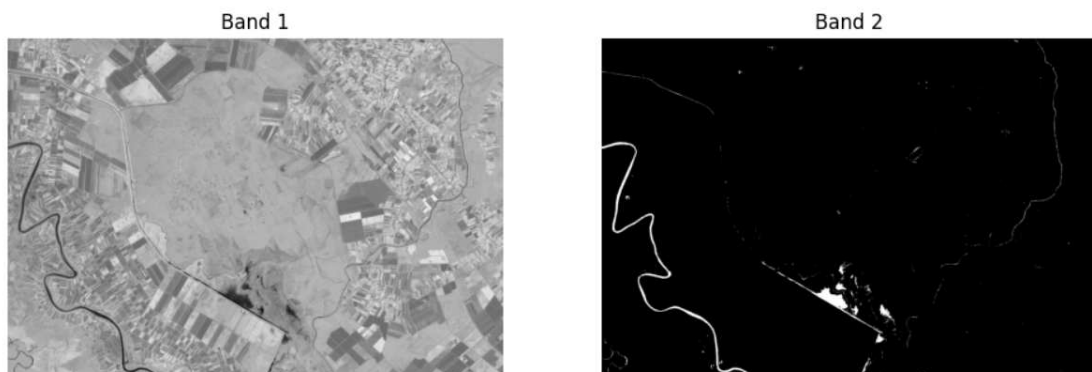
Svaki od spektralnih pojaseva zasebno je prikazan u GeoTIFF formatu [14]. Razlika između klasičnog TIFF formata i dotičnog njegova je mogućnost spremanja metapodataka koji pružaju prostorne informacije. Neke od najvažnijih metapodataka prostorni su opseg i rezolucija, koordinatni referentni sustav (eng. *coordinate reference system*, CRS), broj slojeva u datoteci te matematička pravila za kartografsku projekciju kako bi se podaci iz 3D prostora uspješno transformirali u 2D prikaz. Ovi će podaci biti od iznimne važnosti prilikom pretprocesuiranje snimke opisanom u sljedećem poglavlju.

Kombinacijom različitih pojaseva mogu se izraditi različiti spektralni indeksi koji do-

datno govore o vegetativnom sastavu snimljenog područja. Od šesnaest izrađenih indeksa za ovaj zadatak najzanimljiviji je Normalizirani Diferencijalni Vegetacijski Indeks (eng. *Normalized Difference Vegetation Index*, NDVI). Koristi se kako bi se procijenila količina vegetacije iskorištavajući svojstvo vegetacije da reflektira različite količine svjetlosti u različitim dijelovima spektra. Izračunava se koristeći dva spektralna pojasa (Jednadžba 4.1), crveni (RED) i bliski infracrveni (NIR).

$$NDVI = \frac{(NIR - RED)}{(NIR + RED)} \quad (4.1)$$

Raspon je NDVI vrijednosti od 0 do 1, gdje vrijednosti veće od 0.6 predstavljaju površine s gustom vegetacijom poput šuma i poljoprivrednih područja te se zbog toga koristi u zadatku. Niže vrijednosti predstavljaju travnjake, pustinje i stijene, a vrijednosti približne nuli upućuju na vodene površine. Prikaz jednog indeksa može se vidjeti na slici 4.1., na kojoj jedan sloj predstavlja vegetaciju, a drugi vodenu površinu.



Slika 4.1. Normalizirani Diferencijalni Vegetacijski Indeks

Zbog svoje jednostavnosti i učinkovitosti u procjeni vegetacijske površine NDVI se koristi u raznim područjima poput poljoprivrede, ekologije te klimatologije.

4.2. Segmentacijske maske

Prilikom provođenja zadatka segmentacije koristilo se pet maski koje predstavljaju ispravne podatke (eng. *ground truth*). Oni su ključni pri treningu modela i mijenjanju težina kako bi model naučio prepoznati i klasificirati različite šumske vrste. Također se koriste pri evaluaciji uspoređujući točne vrijednosti s procjenama modela. Zabilježene

su šumske vrste grab, hrast, jasen, joha i vrba te je svakoj vrsti pridružena jedna maska (Slika 4.2.).



Slika 4.2. Prikaz jasenove maske na satelitskoj snimci

Sve se maske odnosi na isto geografsko područje zabilježeno satelitom. Upravo se zbog toga ovakva vrsta segmentacije razlikuje od klasične više klasne segmentacije (eng. *multi-class classification*) jer pojedinačni piksel ne pripada samo jednoj klasi. Za svaki piksel definirana je vjerojatnost pripadnosti određenoj klasi. Takva vrsta segmentacije naziva se višeoznačna klasifikacija (eng. *multi-label classification*). Međutim, zbrojem vjerojatnosti pripadnosti navedenih pet klasa ukupna suma ne iznosi 100% te se ona treba nadopuniti. Time nastaje šesta klasa koja predstavlja ostale šumske vrste o kojima nisu dostupni točni podatci.

5. Metodologija

5.1. Korištene tehnologije

U rješavanju zadatka korišten je programski jezik *Python* (verzija 3.10) zbog svoje široke primjene u razvoju modela dubokog učenja. Koristio se u sklopu usluge *Google Colab* koja se često koristi zbog svoje jednostavnosti i besplatnog pristupa jačim računalnim resursima. Na izbor se također nudi veliki broj biblioteka koje olakšavaju analizu slika, vizualizaciju rezultata te, najbitnije, implementaciju dubokih neuronskih mreža. Uz standardne *Python* biblioteke za razne matematičke operacije i izračune koristila se *NumPy* biblioteka. Pri analizi slike koristila se biblioteka *rasterio* koja je personalizirana za rad s GeoTIFF vrstom podataka. Za vizualizaciju podataka koristile su se biblioteke *Matplotlib* i *Seaborn*. Implementacija modela provodila se pomoću biblioteke *PyTorch*.

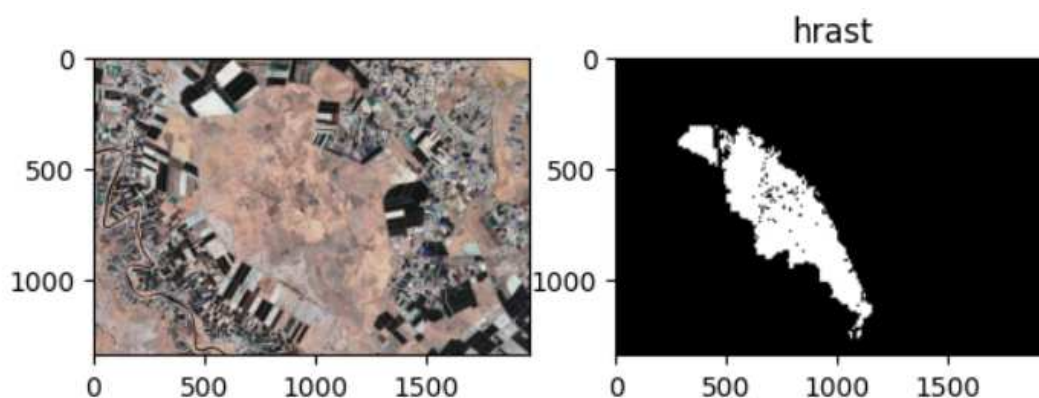
5.2. Pretprocesiranje podataka

Kako bi treniranje modela bilo uspješno, sirove je podatke potrebno pretprocesirati kako bi se zadovoljili svi uvjeti modela. Jedan je od predodređenih uvjeta modela da se na ulaz šalje slika koja sadržava tri kanala odnosno njezine dimenzije su (H, W, 3). U nastavku rada, slovo H označavat će visinu (eng. *height*), a slovo W širinu (eng. *width*) slike. Analizom jedne satelitske snimke dolazi se do informacije da je ona crno-bijela odnosno posjeduje samo jedan kanal. Kako bi se zadovoljio uvjet više kanalne slike, potrebno je odabrati tri različite snimke te ih naslagati jednu na drugu. Tijekom 15 dana u travnju zabilježene su tri snimke tijekom proljetnog cvata. Iz tog su razloga navedene snimke odabrane te se njihovi slojevi u NDVI koji predstavljaju vegetaciju slažu pomoću funkcije *numpy.stack()*. Dobivena višekanalna slika koristit će se u daljnjem pretprocesiranju (Slika 5.1.).



Slika 5.1. Dobivena višekanalna slika

Proučavanjem rezolucije satelitskih snimaka i njihovih pridruženih maski uočava se velika nepodudarnost. Rezolucija snimki iznosi 1967×1342 (H x W), dok je rezolucija maske znatno manja te iznosi 116×114 . Ista rezolucija slike i njezine maske također je jedan od uvjeta modela. Kako bi se zadovoljio taj uvjet, potrebno je iz metapodataka snimke i maske pridobiti informacije o transformaciji koja određuje kako se koordinatni sustav slike preslikava na stvarne koordinate na površini. I snimka i maska koriste *Affine* transformaciju koja je linearna transformacija te uključuje skaliranje, rotaciju te translaciju. Obje također koriste isti CRS koji će nam pomoći pri usklađivanju. Korištenjem funkcije *reproject* iz biblioteke *rasterio* s parametrima spomenutih transformacija i CRS mijenja se rezolucija maske da odgovara rezoluciji snimke (Slika 5.2.) koristeći metodu najbližeg susjeda.



Slika 5.2. Slika i odgovarajuća maska za hrast

Primjenom funkcija iz biblioteke *NumPy* uz pomoć pet predodređenih maski izračuna se prije spomenuta šesta maska koja predstavlja klasu svih ostalih šumskih vrsta. Njezinu je rezoluciju također potrebno izjednačiti s rezolucijom snimke koristeći opisani postupak.

Uspješnost modela uvelike ovisi o veličinu skupa za treniranje odnosno količini podataka. Najkvalitetniji modeli današnjice trenirani su na velikim skupovima podataka kako bi pridobili svojstvo generalizacije. Kako bi se dobio što veći skup, potrebno je sliku i odgovarajuću masku izrezati na slike manje rezolucije. Zbog rada modela izrezani dio trebao bi biti kvadratnog oblika s dimenzijom višekratnika broja 2. Proučavanjem slike 5.2. uočava se da maska ne prekriva cijelu površinu snimke odnosno da na nekim dijelovima nema podataka. Takvi će se dijelovi maske zanemariti pod uvjetom da na izrezanoj masici ne prekrivaju više od 50% njezine površine. Nakon što izrezana maska zadovolji taj uvjet, slika će se također izrezati na dijelove koji odgovaraju poziciji izrezanih maski. Ovim postupkom (Funkcija 5.1) dobivamo skup podataka izrezanih slika i maski koji će se kasnije koristiti pri treningu modela. Za svaku klasu odnosno njenu masku postupak se ne mijenja te se tako osigurava da geografske pozicije maski za različite klase odgovaraju jedna drugoj. Odabirom dimenzije 32 x 32 za novoizrezane slike odnosno maske skup podataka sadržava 236 parova slika i odgovarajućih maski koje zadovoljavaju postavljene uvjete.

Listing 5.1: Funkcija za izrezivanje poligona iz maske

```
1 def cropping_polygons(mask, new_dimension, no_data):
2     valid_coordinates = []
```

```

3 background_threshold = 0.5
4 num_horizontal = mask.shape[1] // new_dimension
5 num_vertical = mask.shape[0] // new_dimension
6
7 for i in range(num_vertical):
8     for j in range(num_horizontal):
9         top_left = (j * new_dimension, i * new_dimension)
10        bottom_right = ((j + 1) * new_dimension, (i + 1) *
11                        new_dimension)
12        small_img = mask[top_left[1]:bottom_right[1], top_left
13                        [0]:bottom_right[0]]
14        background_coverage = np.sum(small_img == no_data) / (
15                                new_dimension * new_dimension)
16
17        if background_coverage < background_threshold:
18            valid_coordinates.append((top_left, bottom_right))
19
20 return valid_coordinates

```

Kako bi se svi podatci mogli ispravno predati na ulaz modela potrebno ih je ispravno reprezentirati. Korištenjem iterabilnog skupa podataka moguće je ispravno prikazati par *slika, maska*. On se implementira pomoću *torch.util.data.Dataset* klase koja se nasljeđuje u implementaciji određenoj za zadatak (Klasa 5.2).

Listing 5.2: Klasa za skup podataka

```

1 class ForestDataset(Dataset):
2     def __init__(self, image, masks, image_transform=None,
3                 mask_transform=None):
4         self.image = image
5         self.masks = masks
6         self.image_transform = image_transform
7         self.mask_transform = mask_transform
8
9     def __len__(self):
10        return len(self.image)

```



```

10
11     def __getitem__(self, idx):
12         image = self.image[idx]
13         masks = {name: mask[idx] for name, mask in self.masks.
14                   items()}
15
16         if self.image_transform:
17             image = self.image_transform(image)
18         if self.mask_transform:
19             masks = {name: self.mask_transform(mask) for name,
20                      mask in masks.items()}
21
22         masks_tensor = torch.stack(list(masks.values()), dim=0)
23
24         / 100
25
26         return image, masks_tensor

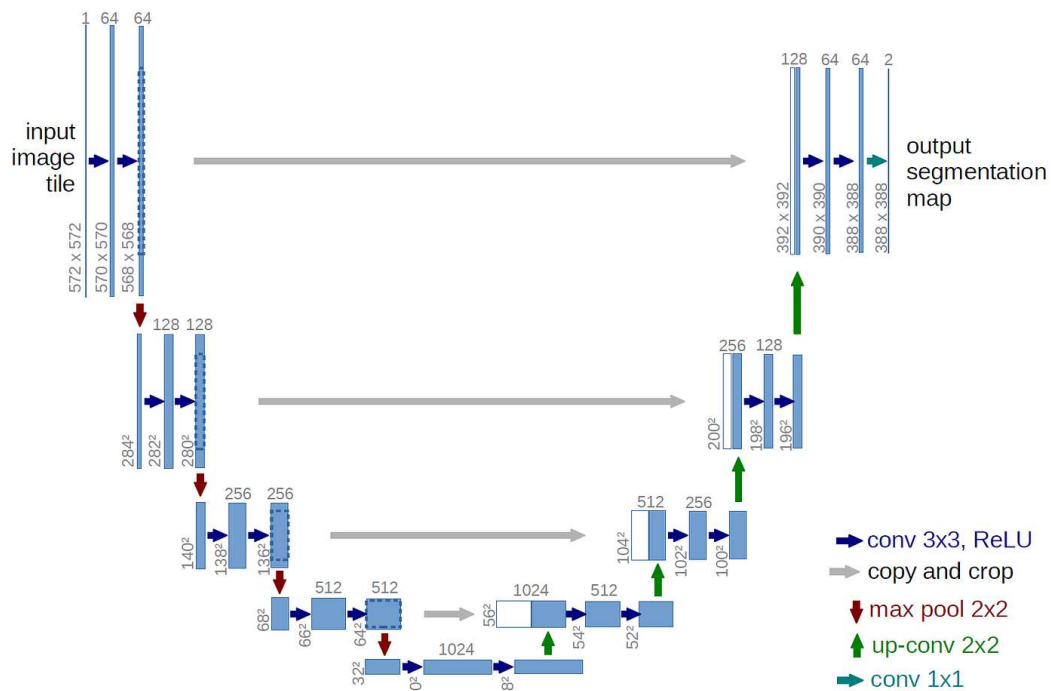
```

Maska je dimenzije (C, H, W) gdje C predstavlja broj klasa prilikom segmentacije. Ovakvim se rasporedom osigurava predaja ispravnih vjerojatnosti za određenu klasu. Posljednji je korak dijeljenje vrijednosti maski sa 100 čime se osigurava interval vjerojatnosti od 0 do 1.

5.3. UNET model

5.3.1. Općenito

UNET model [15] [16] [17] konvolucijska je neuronska mreža razvijena prvobitno za segmentaciju slika u biomedicini, međutim pronalazi primjenu i u drugim zadacima segmentacije. Model je razvio Odjel za računalnu znanost na Sveučilištu u Freiburgu. Svoje prepoznatljivo ime dobio je po svom obliku koji podsjeća na slovo U (Slika 5.3.).



Slika 5.3. UNET model [16]

5.3.2. Arhitektura UNET modela

Model se sastoji od dva glavna dijela, enkodera i dekodera. Enkoder služi kako bi prepolovio rezoluciju slike i udvostručio broj značajki koje se promatraju. Suprotno, dekodер služi kako bi udvostručio rezoluciju, a prepolovio broj promatranih značajki. Zaključno, enkoder služi za ekstrakciju značajki objekta koji se nalaze na slici, a dekodер služi za rekonstrukciju rezolucije slike i precizno lociranje objekata na slici. Obojica se sastoje od četiri bloka gdje je zadnji blok enkodera povezan mostom (eng. *bottleneck*) s prvim blokom dekodera.

Prvi enkoderski blok predstavlja i ulaz u cijelu mrežu kojem se predaje slika na kojoj se želi provesti segmentacija. Slika se zatim propušta kroz dvodimenzionalnu konvoluciju kojoj je filter veličine 3x3. Nakon konvolucije se primjenjuje aktivacijska funkcija zglobnice čime se u mrežu uvodi nelinearnost koja poboljšava generalizaciju nad podacima za trening. Ovaj se postupak ponavlja još jedanput te tako predstavlja konvolucijski sloj enkodera. Nakon njega slijedi maksimalno sažimanje koristeći 2x2 filter čime se rezolucija slike prepolovila pritom zadržavajući ključne karakteristike. Takva se slika predaje sljedećem bloku u nizu.

Često se između konvolucije i aktivacijskog sloja primjenjuje normalizacija (eng. *Batch Normalization*) podataka kojom osiguramo stabilnost pri treningu. Pri poboljšanju generalizacije također se može koristiti ispuštanje nasumičnih piksela (eng. *dropout*) nakon primjene aktivacijske funkcije čime mreža postaje manje ovisna o pojedinim neuronima. Navedene dvije tehnike nisu obavezne pri implementaciji modela međutim uvelike poboljšavaju njegove performanse.

Enkoderski blok: $\text{Conv2D} \rightarrow (\text{Batch Normalization}) \rightarrow \text{ReLU} \rightarrow (\text{Dropout}) \rightarrow \text{Conv2D} \rightarrow (\text{Batch Normalization}) \rightarrow \text{ReLU} \rightarrow (\text{Dropout}) \rightarrow \text{MaxPooling}$

Nakon posljednjeg enkoderskog bloka slijedi prije spomenuti most koji sadrži najviše i najkompleksnije karakteristike slike unatoč najmanjoj dimenziji slike u modelu. Jedina je razlika između njega i enkoderskog bloka izostavljanje maksimalnog sažimanja. Njegova je glavna uloga spoj enkodera i dekodera.

Bottleneck: $\text{Conv2D} \rightarrow (\text{Batch Normalization}) \rightarrow \text{ReLU} \rightarrow (\text{Dropout}) \rightarrow \text{Conv2D} \rightarrow (\text{Batch Normalization}) \rightarrow \text{ReLU} \rightarrow (\text{Dropout})$

Dekoderski blok započinje povećanjem rezolucije (eng. *upsampling*). Kako bi se pridobile dodatne informacije o točnoj lokaciji karakteristika objekta na slici, unutar dekodera koriste se skip-konekcije (eng. *skip-connections*). Skip-konekcija povećanjem rezolucije spaja (eng. *concatenate*) izlaz odgovarajućeg enkoderskog bloka sa slikom. Uz poboljšanje protoka gradijenta tijekom propagacije unatrag, skip-konekcije također pružaju značajke odgovarajućeg enkoderskog bloka koji su se po mogućnosti izgubile zbog dubine mreže. Nakon toga, slika se ponovno propušta kroz dvije konvolucije s aktivacijom.

Dekoderski blok: $\text{Upsampling} \rightarrow \text{Concatenate} \rightarrow \text{Conv2D} \rightarrow (\text{Batch Normalization}) \rightarrow \text{ReLU} \rightarrow (\text{Dropout}) \rightarrow \text{Conv2D} \rightarrow (\text{Batch Normalization}) \rightarrow \text{ReLU} \rightarrow (\text{Dropout})$

Izlaz je cijele mreže konvolucija s filterom 1 x 1 te odgovarajućom aktivacijskom funkcijom. Ako se radi o segmentaciji s dvije klase koristi se sigmoidalna funkcija, a s više se klasa koristi *softmax* funkcija. Primjenom aktivacijske funkcije stvara se predviđena segmentacijska maska za svaki piksel na slici.

Za implementaciju cijelog modela koristila se biblioteka *PyTorch* odnosno njen mo-

dul *torch.nn* koji sadržava raznolike klase koje pomažu pri implementaciji modela neuronskih mreža. Korištenje modula prikazano je klasom 5.3) u kojoj se implementira dvostruka konvolucija s filterom 3x3 odnosno konvolucijski sloj koji se koristi tijekom cijele mreže. Konvolucija koja se koristila nije mijenjala rezoluciju ulazne slike.

Listing 5.3: Konvolucijski sloj

```
1 class DoubleConv(nn.Module):
2     def __init__(self, in_channels, out_channels):
3         super(DoubleConv, self).__init__()
4         self.conv = nn.Sequential(
5             nn.Conv2d(in_channels, out_channels, kernel_size = 3,
6                       padding = 1),
7             nn.BatchNorm2d(out_channels),
8             nn.ReLU(inplace = True),
9             nn.Conv2d(out_channels, out_channels, kernel_size = 3,
10                      padding = 1),
11             nn.BatchNorm2d(out_channels),
12             nn.ReLU(inplace = True),
13         )
14     def forward(self, x):
15         return self.conv(x)
```

5.4. Trening i evaluacija

Prilikom treniranja i evaluacije UNET modela početni skup podataka od 236 slika raspodijeljen je u tri skupa. Skupu za trening nasumično je pridijeljeno 165 slika, dok se skup za validaciju sastoji od 47 slika. Ostalih 24 slike je pripalo skupu za testiranje.

5.4.1. Hiperparametri modela

Tijekom treniranja modela koristili su se razni hiperparametri koji služe za poboljšanje uspješnosti modela. Hiperparametri su parametri čije se vrijednosti postavljaju prije treninga i ne mijenjaju se tijekom cijelog procesa. Korišteni su hiperparametri: broj epoha, stopa učenja (eng. *learning rate*), veličina grupa (eng. *batch size*), težina klase, parametar propadanje težine (eng. *weight decay*).

Broj epoha određuje koliko će puta cijeli skup za trening biti korišten kako bi se optimizirale težine modela. Veći broj epoha uzrokuje dulji vremenski tijek treninga, međutim najčešće uzrokuje i poboljšanje modela ako se pravilno izbjegne prenaučенost. Stopa je učenja vrijednost koja kontrolira veličinu koraka koju optimizator može napraviti tijekom izmjene težine. Veličina grupa određuje broj uzoraka iz skupa podataka koji se koriste pri izračunavanju gradijenta u jednom koraku treniranja. Manjim brojem uzoraka poboljšava se generalizacija i brzina treninga, ali to utječe na njegovu stabilnost. Suprotno, veći broj uzoraka pridonosi stabilnosti, ali usporava trening jer zahtijeva više memorije.

Navedena tri parametra ključna su za ispravan trening i gotovo se uvijek definiraju. Težina klasa nije jedan od obaveznih hiperparametara, ali služi kako bi uravnotežili doprinos različitih klasa tijekom izračuna gubitka zbog njihove neuravnoteženosti. Parametar propadanja težine koristi se kako bi se izbjeglo moguće prekomjerna prilagodba modela trening podatci čime se izbjegava prenaučенost. Točne korištene vrijednosti hiperparametara iznose:

- broj epoha: maksimalno 100
- stope učenja: 1×10^{-5} , 5×10^{-5} , 1×10^{-4} , 5×10^{-4} , 1×10^{-3} , 5×10^{-3} , 1×10^{-2} , 5×10^{-2}
- veličina grupe: 4
- težina klasa:
 - vrba: 0.2445
 - hrast: 0.0719
 - jasen: 0.0745
 - grab: 0.2408
 - joha: 0.3497
 - ostale šumske vrste: 0.0186

- vrijednost propadanje težine: 1×10^{-5}

Uvođenje maksimalnog broja epoha od 100 koristi se kako bi se spriječilo prekomjerno treniranje modela, što može dovesti do prenaučenosti. Rano zasutavljanje (eng. *Early stopping*) tehnika prati grešku modela na validacijskom skupu tijekom treninga i zaustavlja trening ako se greška prestaje smanjivati, čime se dodatno osigurava protiv prenaučenosti. Model će zaustaviti trening ako se nakon 5 uzastopnih epoha greška dodatno ne smanji. Za postizanje optimalnih rezultata, model će se testirati s različitim stopama učenja: 1×10^{-5} , 5×10^{-5} , 1×10^{-4} , 5×10^{-4} , 1×10^{-3} , 5×10^{-3} , 1×10^{-2} , i 5×10^{-2} . Svaka od stopa učenja će biti isprobana kako bi se pronašla ona koja najbolje konvergira za određeni problem. Na taj način, kombinacija ranog zaustavljanja i prilagodbe stope učenja omogućava najbolju izgradnju modela.

5.4.2. Funkcija greške i optimizator

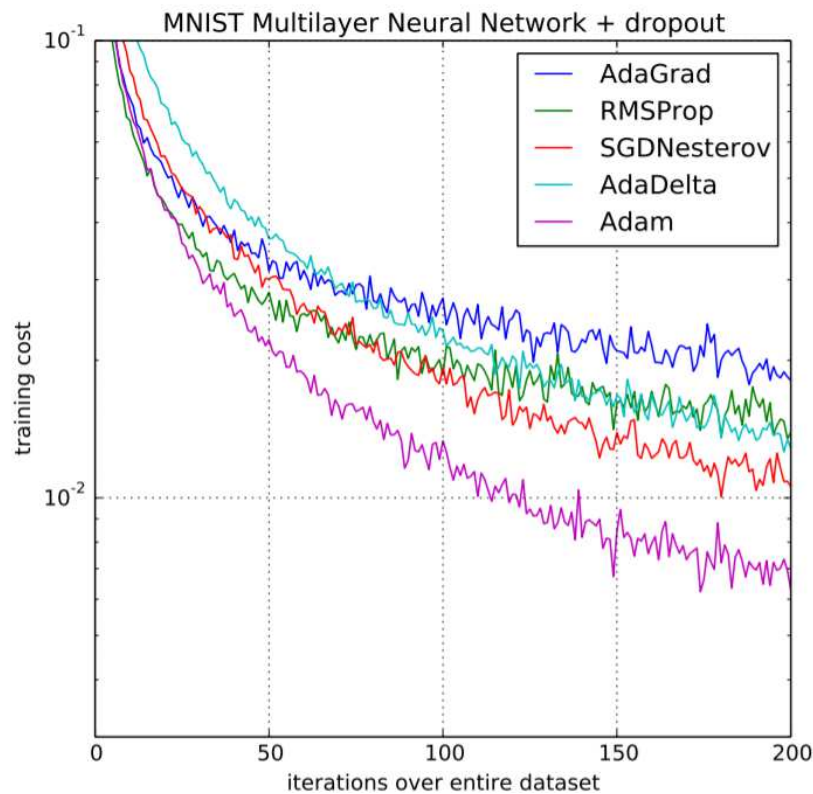
Funkcija greške izračunava skalarnu vrijednost koja predstavlja razliku između stvarne vrijednosti i predikcije modela. Kako se u zadatku koriste vjerojatnosti za različite klase umjesto pripadnosti piksela samo jednoj klasi, mora se koristiti binarna unakrsna entropija (eng. *Binary Cross-Entropy*, BCE) kao funkcija greške za svaki pojedinačni element u vektor (Jednadžba 5.1). U računanju pogreške korišteni su samo pikseli koji sadržavaju vrijednost odnosno pozadina koja nema određenu vrijednost nije utjecala na grešku kao ni na učenje modela.

$$\text{BCE} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (5.1)$$

- y_i : stvarne vrijednosti (vjerojatnosti)
- p_i : predikcije modela (vjerojatnosti)

Optimizator je algoritam koji koristi gradijente funkcije greške kako bi mijenjao težine modela s ciljem minimiziranja greške. Gradijenti su izračunati u procesu propagacije unatrag. U rješavanju ovog problema koristio se optimizator Adam (eng. *Adaptive Moment Estimation*) [18]. Pri svojoj implementaciji koristi dijelove implementacije optimizatora RMSProp (eng. *Root Square Mean Propagation*) i stohastički gradijentni spust

(eng. *Stochastic Gradient Descent*, SGD). Prednost RMSProp-a koja se koristi prilagodljive su stope učenja koje imaju osnovu na prosjecima kvadrata gradijenta koji se ujedno koriste u SGD za mijenjanje težina. Njegovom uporabom omogućen je efikasniji i stabilniji trening s obzirom na druge optimizatore (Slika 5.4.).



Slika 5.4. Usporedba optimizatora Adam s drugima [18]

5.4.3. Dodatna poboljšanja

Dodatni alat koji pomaže pri ubrzanju treninga gradijentni je podešivač (eng. *Grad-Scaler*). Podešivač se koristi na paralelnoj računalnoj platformi CUDA koju je razvila tvrtka NVIDIA koja koristi grafičke programske jedinice (*GPU*) kako bi pomogla u kompleksnim izračunima. Koristeći mješovitu preciznost s kombinacijom 16-bitne i 32-bitne preciznosti sprječavaju se problemi s preciznošću. Premala promjena težine jedan je od problema koji se često pojavljuje. Sprječavanjem takvih problema osigurava se stabilnost tijekom treninga čime se postiže i efikasnije korištenje dostupnih resursa.

Zbog male veličine ukupnog skupa podataka pri povećanju raznolikosti, koriste se augmentacije. One primjenom različitih transformacija na originalne podatke stvaraju nove promijenjene podatke. Tim postupkom, iako nismo kvantitativno povećali skup

podataka, stvaramo različite primjere iz postojećih podataka. Njihova primjena poboljšava i generalizaciju modela na neviđenim podatcima i smanjuje rizik od prenaučivosti. Neke su od najčešće korištenih transformacija geometrijske transformacije kao što su: rotacija, translacija, skaliranje i zrcaljenje. Uz njih, moguće je promijeniti i svjetlinu i kontrast u fotografiji te dodati šum čime se simuliraju svi mogući uvjeti snimanja.

5.4.4. Evaluacijske metrike

Evaluacijske metrike od iznimne su važnosti za procjenu točnosti modela. U kontekstu ovog zadatka koriste se četiri metrike, Diceov koeficijent, srednja kvadratna pogreška (eng. *Mean Squared Error*, MSE), srednja apsolutna relativna pogreška (eng. *Mean Absolute Relative Error*, MARE) i točnost za pojedinu klasu (eng. *class-wise accuracy*).

Diceov koeficijent ili F1 rezultat koristi se za mjerenje sličnosti između predikcija i stvarnih vrijednosti. Njegove se vrijednosti u intervalu od 0 do 1, gdje koeficijent jednak 1 predstavlja savršeno preklapanje, a 0 predstavlja nikakvo preklapanje. Koeficijent se inače računa dijeljenjem dvostrukog broja piksela u presjeku predikcije i stvarnih vrijednosti sa sumom svih piksela u oba segmenta. Međutim, zbog uporabe kontinuiranih vrijednosti, brojnik računamo kao dvostruki umnožak stvarnih vrijednosti i predikcije, a nazivnik kao zbroj kvadrata stvarnih vrijednosti i kvadrata predikcija modela za sve uzorke (Jednadžba 5.2).

$$\text{Dice} = \frac{2 \sum_{i=1}^N T_i P_i}{\sum_{i=1}^N T_i^2 + \sum_{i=1}^N P_i^2} \quad (5.2)$$

- T_i - stvarna vrijednost za uzorak i
- P_i - predikcija modela za uzorak i
- N - ukupan broj uzoraka

MSE mjeri prosječnu kvadratnu pogrešku između stvarnih vrijednosti i predikcije. Za jednu se sliku MSE računa sumom kvadrata razlike između stvarne vrijednosti i predikcije za svaki piksel. Izračunata se vrijednost zatim podijeli s ukupnim brojem piksela. Veće vrijednosti ukazuju na velike razlike između dva skupa vrijednosti dok sve veća konvergencija prema nuli predstavlja bolju predikciju (Jednadžba 5.3).

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (5.3)$$

- N : Ukupan broj piksela
- y_i : Stvarna vrijednost piksela i
- \hat{y}_i : Vrijednost predikcije piksela i

MARE mjeri prosječnu apsolutnu relativnu pogrešku između stvarnih vrijednosti i predikcije. Za jednu sliku MARE se računa sumom apsolutnih relativnih razlika između stvarne vrijednosti i predikcije za svaki piksel. Izračunata se vrijednost zatim podijeli s ukupnim brojem piksela. Veće vrijednosti ukazuju na velike relativne razlike između dva skupa vrijednosti, dok sve veća konvergencija prema nuli predstavlja bolju predikciju (Jednadžba 5.4).

$$\text{MARE} = \frac{1}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{|y_i|} \quad (5.4)$$

- N : Ukupan broj piksela
- y_i : Stvarna vrijednost piksela i
- \hat{y}_i : Vrijednost predikcije piksela i

Evaluacija modela po klasama omogućava procjenu točnosti predikcija za svaku pojedinačnu klasu, uzimajući u obzir toleranciju odstupanja. Zbog prirode ovog problema odnosno računanja s postotcima uvodi se tolerancija odstupanja od prave vrijednosti u iznosu od 0.085. Ova metrika korisna je pri potrebi ocjenjivanja točnosti modela po klasama, osobito kada su ciljne vrijednosti kontinuirane te je prihvatljivo odstupanje u predikcijama.

Točnost za klasu i može se izraziti sljedećom formulom:

$$\text{Točnost}_i = \frac{\sum_{j=1}^N \mathbb{1}\{\text{donja_granica}_i \leq \text{preds}_{i,j} \leq \text{gornja_granica}_i\}}{N} \quad (5.5)$$

- $\text{preds}_{i,j}$: predikcija za klasu i i uzorak j ,

- $\text{targets}_{i,j}$: prava vrijednost za klasu i i uzorak j ,
- $\text{donja_granica}_i = \text{targets}_{i,j} - \text{odstupanje}$,
- $\text{gornja_granica}_i = \text{targets}_{i,j} + \text{odstupanje}$,
- N : ukupan broj uzoraka za klasu i ,

Ovaj pristup omogućuje detaljnu analizu performansi modela po klasama, što može biti korisno za identificiranje specifičnih područja gdje model treba poboljšanje. Uvođenjem tolerancije odstupanja osigurava se prilagodljivija procjena točnosti, posebno kada su manje pogreške u predikcijama prihvatljive.

6. Rezultati

6.1. Prikaz i rasprava o rezultatima modela

Promjenom hiperparametra stope učenja uvelike se utječe na rezultate modela. Premala stopa učenja može dovesti do dugog vremena treninga, dok prevelika stopa učenja može uzrokovati da model preskoči optimalna rješenja. Evaluacija performansi modela na testnom skupu podataka za različite stope učenja vrši se pomoću prije spomenutih evaluacijskih metrika i vrijednosti funkcije greške (Tablica 6.1.). Manja vrijednost funkcije greške upućuje na bolje performanse modela.

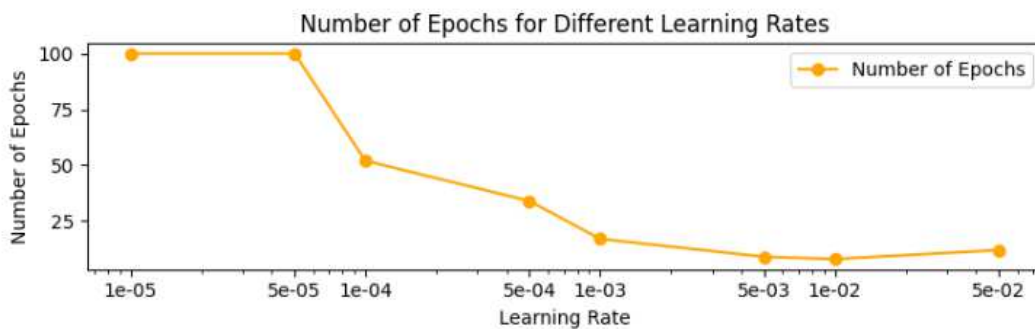
Stopa učenja	Vrijednost funkcije greške	MSE	Dice koeficijent	MARE
1×10^{-5}	0.0539	0.0337	0.8191	81.11
5×10^{-5}	0.0337	0.0147	0.9001	14.54
1×10^{-4}	0.0337	0.0139	0.9054	13.95
5×10^{-4}	0.0336	0.0138	0.9055	12.74
1×10^{-3}	0.0339	0.0138	0.9063	13.73
5×10^{-3}	0.0343	0.0139	0.9071	15.37
1×10^{-2}	0.0344	0.0143	0.9031	14.86
5×10^{-2}	0.0348	0.0149	0.9029	12.12

Tablica 6.1. Rezultati testiranja za različite stope učenja

Primijećuje se da najbolji rezultati model postiže za stope učenja s vrijednošću 5×10^{-4} i 5×10^{-3} , gdje su vrijednosti funkcije greške, MSE i MARE najniže, a Dice koeficijent najviši. One omogućuju modelu da brže i stabilnije konvergira prema optimalnom rješenju. Jedina stopa učenja koja nije zadovoljila s rezultatima je 1×10^{-5} , koja zbog svoje niske vrijednosti nije uspjela konvergirati prema najboljem rješenju.

Dodatna usporedba različitih stopa učenja se može prikazati i preko drugog hiperparametra, broja epoha. Upotrebom ranog zaustavljanja moguće je da se broj epoha za različite stope učenja razlikuje (Slika 6.1.). Manji broj epoha ukazuje na sporiju ko-

nvergaciju prema optimalnom rješenju uz duže trajanje treninga. Suprotno, veći broj epoha brže konvergira prema optimalnom rješenju uz skraćeno vrijeme treninga.



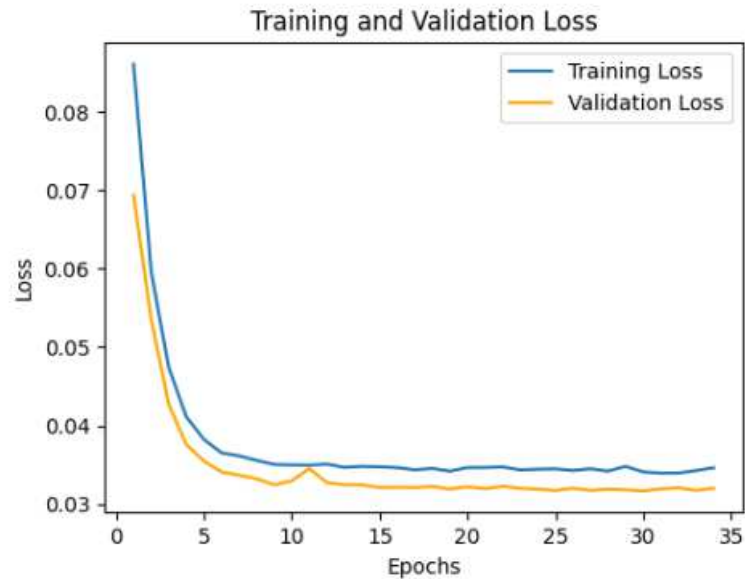
Slika 6.1. Usporedba broja epoha za različite stope učenja

Modeli s dvije najmanje vrijednosti stope učenja dostigle su tijekom treninga maksimalan broj epoha učenja što upućuje na sporiju konvergenciju. Povećanjem broja epoha za model s najmanjom stopom učenja, koji je postigao najgore rezultate, trening se može dodatno poboljšati uz njegovo duže trajanje. Kod modela s najvećim stopama učenja broj epoha treniranja iznosi oko 10 epoha čime se ušteduje na vremenu i računalnim resursima pritom postizujući slične rezultate. Model s vrijednošću stope učenja 5×10^{-4} , koji također pokazuje zavidne rezultate, trening izvršava u 34 epohe.

Detaljnom analizom rezultata pomaže se u odabiru optimalne stope učenja koja omogućava postizanje najboljih performansi modela na testnom skupu podataka. Stoga je važno testirati i usporediti različite stope učenja kako bi se osigurala najbolja moguća generalizacija modela na neviđene podatke.

Performanse modela mogu se prikazati i s pomoću performansi treninga odnosno poboljšanjem ili pogoršavanjem metrika u njegovom trajanju. Sljedeća analiza izvršena je na modelu sa stopom učenja 5×10^{-4} koji je na testnom skupu pokazao uspješne rezultate. Uspješnost treninga i evaluacije najčešće se prikazuje pomoću usporedbe vrijednosti funkcije greške na skupu podataka za trening i validaciju (Slika 6.2.). Postupno smanjenje greške tijekom epoha i treninga i validacije upućuje na vrlo dobro prilagođenje modela podacima pritom izbjegavajući prenaučenosť. Greška se na validacijskom skupu smanjuje s početne vrijednosti 0.0798 na 0.0341. Smanjenje predstavlja sposobnost modela da minimizira pogreške na neviđenim podacima što je ključan element

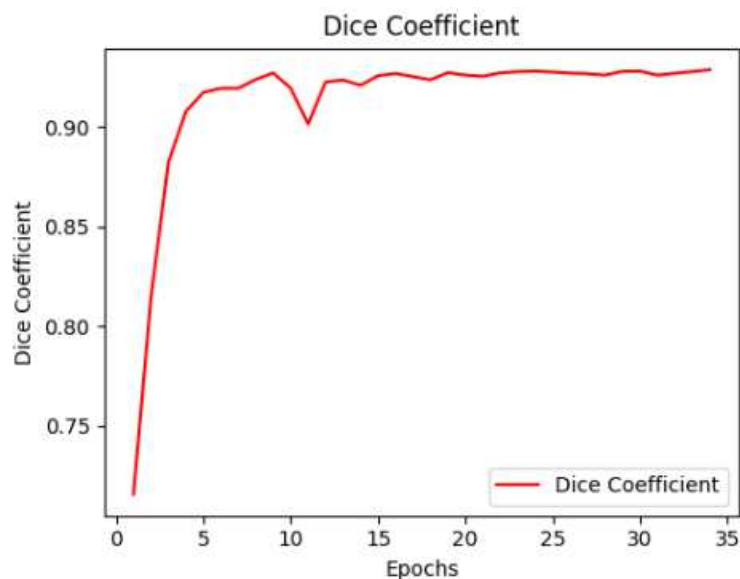
generalizacije.



Slika 6.2. Usporedba greške treninga i validacije

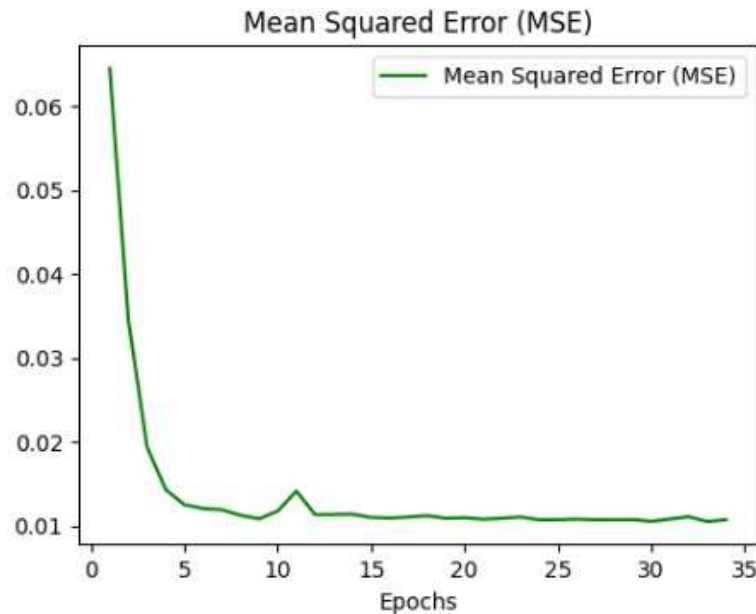
Napredak tijekom treninga također praćen je s pomoću prethodno navedenih evaluacijskih metrika koje su primijenjene na validacijski skup. Općenito, rast Dice koeficijenta te pad srednje kvadratne pogreške i srednje apsolutne relativne pogreške ukazuje na postupno učenje modela kroz svaku epohu.

Dice koeficijent se poboljšao s početne vrijednost 0.7312 na 0.9234 (Slika 6.3.). Konvergencija Dice koeficijenta prema 1 upućuje na uspješan i stabilan trening.



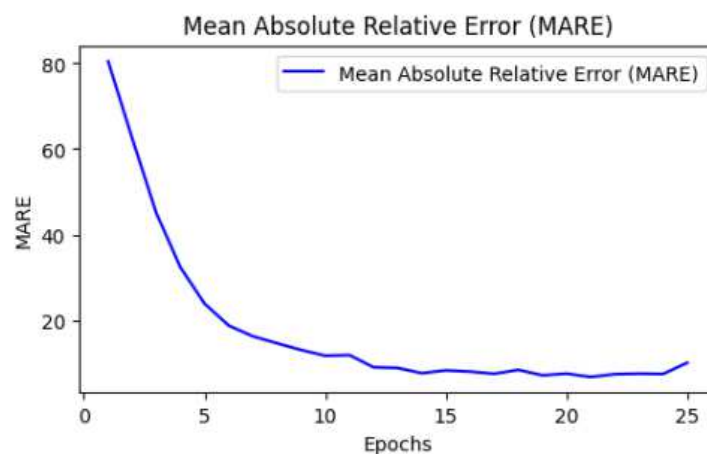
Slika 6.3. Dice koeficijent - stopa učenja 5×10^{-4}

MSE se smanjio s 0.0641 na zavidnih 0.0117 ističući veliko poboljšanje točnosti modela (Slika 6.4.). Konvergencija MSE prema 0 na validacijskom skupu upućuje također na uspješan trening pritom izbjegavajući prenaučenost.



Slika 6.4. Srednja kvadratna pogreška - stopa učenja 5×10^{-4}

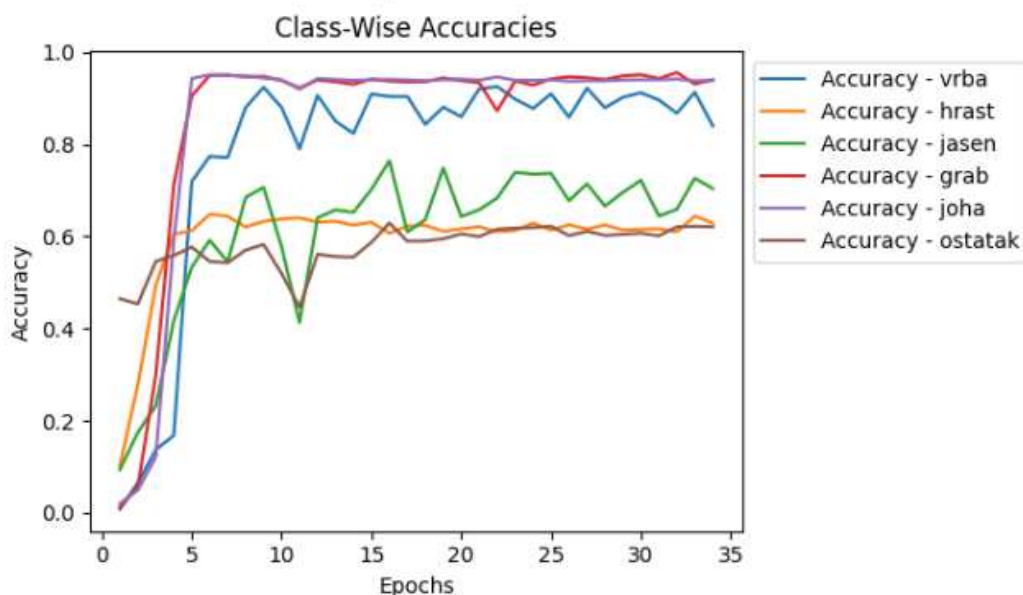
MARE vrijednost se također smanjila s vrijednosti 80.67 na 16.17 te iako je riječ o konvergenciji prema nižim vrijednost, model nije pokazao najbolje rezultate (Slika 6.4.). Mogući problem je računanje s kontinuiranim vrijednostima koje su niže od 1 te one uvelike utječu na krajnji rezultat zbog njihove uloge u nazivniku.



Slika 6.5. Srednja apsolutna relativna pogreška - stopa učenja 5×10^{-4}

Točnost predikcije pojedine klase je evaluacijska metrika koja se nije koristila pri us-

poredbi stopa učenja. Međutim, njene vrijednosti na ovome modelu su slične vrijednostima i na ostalim modelima zbog njihovih srodnih rezultata. Za svaku klasu izračunata je točnost predikcije pritom koristeći toleranciju odstupanja zbog korištenja kontinuiranih vrijednosti (Slika 6.6.).



Slika 6.6. Točnost predikcije pojedine klase - stopa učenja 5×10^{-4}

Izračun točnosti, srednje kvadratne pogreške i srednje apsolutne relativne pogreške za pojedinu klasu također je proveden i na testnom skupu podataka (Tablica 6.2.).

Klasa	Točnost predikcije	MSE	MARE
Vrba	0.8095	0.0243	13.67
Hrast	0.7152	0.0163	18.87
Jasen	0.5803	0.0256	7.71
Grab	0.9797	0.0057	14.93
Joha	0.9588	0.0012	20.49
Ostale šumske vrste	0.5632	0.0240	0.81

Tablica 6.2. Točnost predikcije, MSE i MARE pojedine klase - stopa učenja 5×10^{-4}

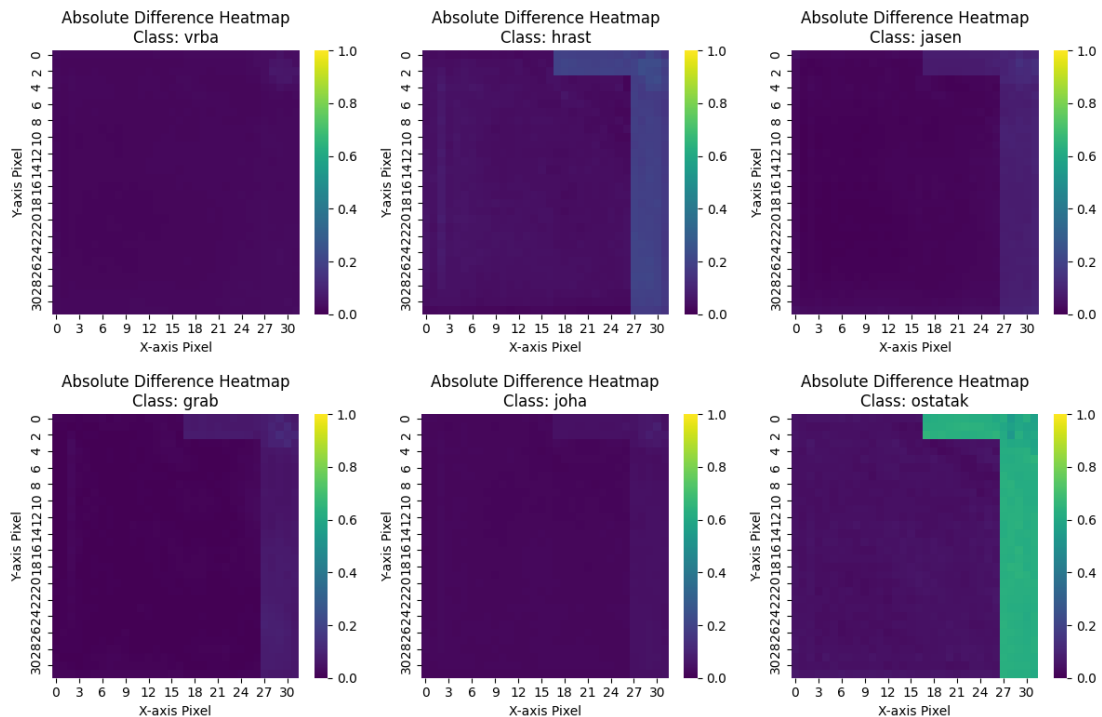
Proučavanjem evaluacijskih metrika za pojedine klase, može se napraviti podjela u tri skupine. Prvu skupinu čine jasen i ostale šumske vrste čija točnost predikcije iznosi oko 60%. Obje vrste su ujedno imale i najveće iznose sa srednju kvadratnu pogrešku. Kod MARE uočavamo anomaliju odnosno spomenute vrste imaju uvelike najmanju pogrešku. Takvi rezultati mogu proizaći iz njihove velike zastupljenosti. Međutim, za ove

dvije vrste model nije najpouzdaniji te se najvjerojatnije ne bi koristio u svrhe njihove predikcije.

Sljedeću skupine čine hrast i vrba. Njihova točnost predikcije je veća te iznosi 70-80%. Vrijednosti srednje kvadratne pogreške su malo manje nego li kod prethodnih dviju vrsta. MARE vrijednost je također visoka kao i za svaku navedenu vrstu. Budući da je hrast druga najzastupljenija vrsta na satelitskim snimkama, njegova točnost predikcije je i više nego prihvatljiva. Model za navedenu skupinu donosi zadovoljavajuće rezultate te bi se mogao koristiti u produkciju uz manju dozu opreza.

Treću skupinu, ujedno i skupinu s najboljem rezultatima, čine grab i joha. Iako je njihova zastupljenost na snimka među najmanjim postotcima, njihova točnost predikcije iznosi preko 95%. Srednja apsolutna pogreška je značajnije manja s obzirom na sve ostale klase. Pogreška je čak deset puta manja nego ostale čime se upućuje na veliku preciznost. Velika MARE vrijednost opravada njihovu nisku zastupljenost u biosustavu. Zbog visoke točnosti i niske zastupljenosti primjenom modela na razne snimke možemo očuvati bioraznolikost prirode.

Kao i u svakoj implementaciji modela, nakon treninga slijedi primjena neuronske mreže na nikad neviđene podatke iz testnog skupa podataka. Dobiveni rezultati se često vizualiziraju uz pomoć originalne slike te odgovarajuće stvarne maske odnosno predikcije. Međutim, budući da svaki piksel na originalnoj slici može pripadati bilo kojoj klasi vizualizacija rezultata je provedena koristeći toplinsku kartu (eng. *heatmap*). Ona prikazuje apsolutnu razliku stvarne vrijednosti i predikcije za svaki piksel pojedine klasu na odabranoj slici (Slika 6.7.). Područja s tamnijim nijansama ukazuju da su predviđanja vrlo slična stvarnim vrijednostima.



Slika 6.7. Toplinska karta slike iz testnog skupa podataka

Proučavanjem toplinske karte uočava se prevladavanje upravo tamnijih nijansa kojima se dokazuje preciznost modela i na nikad viđenim podacima. Područja u kojima uočavaju svjetlije nijanse upućuju da se na njemu ne nalaze nikakvi podaci odnosno prikazana je pozadina slike. Uz vizualizaciju, još jedan način ispitivanja preciznosti modela je izračunavanje metrika na testnom skupu. Njihove vrijednost su:

- Vrijednost funkcije greške na testnom skupu: 0.0336
- Srednja kvadratna pogreška (MSE): 0.0138
- Srednja apsolutna relativna pogreška (MARE): 12.75
- Dice koeficijent: 0.9055

Navedeni rezultati, isključujući srednju apsolutnu relativnu pogrešku, pokazuju da je model naučio izvršiti zadatak odnosno analizirati i segmentirati šumske vrste iz satelitskih snimaka. Procesom treniranja dobio se robusni model sa svojstvom generalizacije. Lako je moguće da bi model imao i bolje performanse ako bi se koristio veći skup podataka. Manji skupovi, pogotovo dotični veličine 236 slika, uvelike utječu na trening po-

većavajući mogućnost prenaučivosti. Upravo zbog toga, odabir UNET modela za takve tipove problema je opravdan što potvrđuju i prikazani rezultati.

7. Zaključak

Istraživanje predstavljeno u ovom radu ima naglasak na primjenu UNET modela za segmentaciju i klasifikaciju šumskih vrsta s pomoću satelitskih snimaka *Sentinel-2*. Cilj je bio razviti učinkovit model koji može precizno klasificirati različite vrste vegetacije u šumskim predjelima. UNET model odabran je zbog svoje dokazane učinkovitosti s manjim skupovima podataka i zadacima segmentacije slika u različitim područjima uključujući biomedicinu i analizu satelitskih snimaka.

Model je treniran na skupu podataka koji se sastoji od 236 slika primjenjujući različite stope učenja, a evaluiran je i testiran korištenjem različitih evaluacijskih metrika kao što su srednja kvadratna pogreška (MSE), Dice koeficijent i točnost predikcije za pojedinu klasu. Rezultati pokazuju da modeli sa vrijednošću stope učenja 5×10^{-4} i 5×10^{-3} postižu najbolje rezultate. Njihova visoka preciznost, s MSE vrijednošću oko 0.01, i Dice koeficijentom većim od 0.9, potvrđuje sposobnost modela da izvrši ispravnu segmentaciju i klasifikaciju šumskih vrsta. Najbolje predviđenje šumske vrste su grab i joha, dok su najlošije jasen i ostale šumske vrste koje nisu označene na satelitskim snimkama. Vizualizacija rezultata uz pomoć toplinske karte i točnosti predikcije za pojedinu klasu dodatno demonstriraju preciznost predikcije modela na nikad viđenim podacima.

Znanstveni je doprinos ovog rada u uspješnoj primjeni dubokih neuronskih mreža prilikom analize ekoloških podataka što omogućuje bolje razumijevanje i praćenje promjena u šumskim predjelima. Praktičnim doprinosom smatra se razvoj alata koji može pomoći u zaštiti bioraznolikosti, očuvanju prirodnih staništa i održivom upravljanju šumama.

Buduća istraživanja trebala bi se usredotočiti na proširenje skupa podataka kako bi se smanjila mogućnost prenaučivosti modela i dodatno poboljšala njegova točnost. Tako-

der, u istraživanje je moguće uključiti primjenu razvijenog modela na različitim geografskim područjima i korištenje naprednijih tehnika pretprocesuiranja podataka kako bi se povećala efikasnost modela. Primjenom drugih modela za segmentaciju u budućim istraživanjima moguće je usporediti rezultate odnosno doći do procjene najboljeg modela za segmentaciju šumskih vrsta.

Literatura

- [1] Y. Chintaram, “Lane detection with semantic segmentation”, <https://github.com/yudhisteer/Lane-Detection-with-Semantic-Segmentation>, [mrežno; stranica posjećena: lipanj 2024.].
- [2] IBM (International Business Machines), <https://www.ibm.com/topics/semantic-segmentation>, [mrežno; stranica posjećena: lipanj 2024.].
- [3] ESA (European Space Agency), <https://sentiwiki.copernicus.eu/web/sentiwiki>, [mrežno; stranica posjećena: lipanj 2024.].
- [4] B. D. Bašić, M. Čupić, i J. Šnajder, “Umjetne neuronske mreže”, <http://java.zemris.fer.hr/nastava/ui/>, 2019/2020.
- [5] Wikipedia contributors, “Umjetna neuronska mreža — wikipedia, the free encyclopedia”, https://hr.wikipedia.org/wiki/Umjetna_neuronska_mre%C5%BEEa, [mrežno; stranica posjećena: lipanj 2024.].
- [6] S. A. Stel'makh, E. M. Shcherban', A. N. Beskopylny, L. R. Mailyan, B. Meskhi, I. Razveeva, A. Kozhakin, i N. Beskopylny, “Prediction of mechanical properties of highly functional lightweight fiber-reinforced concrete based on deep neural network and ensemble regression trees methods”, *Materials*, sv. 15, br. 19, str. 6740, listopad 2022., [mrežno; stranica posjećena: lipanj 2024.]. <https://doi.org/10.3390/ma15196740>
- [7] N. Gupta, “A guide to supervised learning”, <https://medium.com/@ngneha090/a-guide-to-supervised-learning-f2ddf1018ee0>, [mrežno; stranica posjećena: lipanj 2024.].

- [8] A. Vasiliev, “Mathematics behind basic feed-forward neural network (3 layers, python, from scratch)”, <https://medium.com/swlh/mathematics-behind-basic-feed-forward-neural-network-3-layers-python-from-scratch-df88085c8049>, [mrežno; stranica posjećena: lipanj 2024.].
- [9] Wikipedia contributors, “Convolution, wikipedia, the free encyclopedia”, <https://en.wikipedia.org/wiki/Convolution>, [mrežno; stranica posjećena: lipanj 2024.].
- [10] T. D. Science, “Understand the architecture of cnn”, <https://towardsdatascience.com/understand-the-architecture-of-cnn-90a25e244c7>, [mrežno; stranica posjećena: lipanj 2024.].
- [11] S. Team, “Convolutional neural networks: 1998-2023 overview”, <https://www.superannotate.com/blog/guide-to-convolutional-neural-networks>, [mrežno; stranica posjećena: lipanj 2024.].
- [12] I. Team, “Fully connected layers in convolutional neural networks”, <https://indiantechwarrior.com/fully-connected-layers-in-convolutional-neural-networks/>, [mrežno; stranica posjećena: lipanj 2024.].
- [13] D. B. Team, “Fully connected layers in convolutional neural networks”, <https://developersbreach.com>, [mrežno; stranica posjećena: lipanj 2024.].
- [14] H. Team, “What is geotiff? definition and faqs”, <https://www.heavy.ai/technical-glossary/geotiff>, [mrežno; stranica posjećena: lipanj 2024.].
- [15] O. Ronneberger, P. Fischer, i T. Brox, “U-net: Convolutional networks for biomedical image segmentation”, <http://arxiv.org/abs/1505.04597>, 5 2015., [mrežno; stranica posjećena: lipanj 2024.].
- [16] A. Vidhya, “What is unet?” <https://medium.com/analytics-vidhya/what-is-unet-157314c87634>, [mrežno; stranica posjećena: lipanj 2024.].
- [17] L. Dikić, I. Mihaljević, I. Kovač, D. Poljak, i M. Jakovac, “Segmentacija tumora jetre”, 2024.

- [18] G. Team, “Adam optimizer”, <https://www.geeksforgeeks.org/adam-optimizer/>,
[mrežno; stranica posjećena: lipanj 2024.].

Sažetak

Analiza satelitskih snimaka za klasifikaciju šumskih vrsta

Matija Jakovac

Rad istražuje primjenu UNET modela odnosno konvolucijske neuronske mreže za segmentaciju i klasifikaciju šumskih vrsta koristeći satelitske snimke *Sentinel-2*. Model je treniran na skupu podataka od 236 slika, a evaluiran korištenjem metrika poput srednje kvadratne pogreške (MSE), srednje apsolutne pogreške (MARE) i Dice koeficijenta. Uspješan trening postignut je izborom pravilnih hiperparametra. Postignuti rezultati na testnom skupu podataka pokazuju visoku preciznost modela, s MSE od 0.0138, Dice koeficijentom od 0.9055 i točnosti predikcije za grab i johu većim od 0.95. Vizualizacija rezultata toplinskom kartom i raznim grafovima potvrđuje visoku točnost predikcije na neviđenim podacima. Ovaj rad doprinosi razvoju učinkovitog alata za analizu ekoloških podataka te je s njime moguće pomoći u zaštiti bioraznolikosti i održivom upravljanju šumskim resursima.

Ključne riječi: segmentacija; klasifikacija; satelit *Sentinel-2*; konvolucijske neuronske mreže; UNET model; srednja kvadratna pogreška; srednja apsolutna relativna pogreška; dice koeficijent; točnost predikcije

Abstract

Satellite image analysis for classification of forest species

Matija Jakovac

The paper investigates the application of the UNET model, that is, the convolutional neural network for the segmentation and classification of forest species using *Sentinel-2* satellite images. The model was trained on a dataset of 236 images, and evaluated using metrics such as mean square error (MSE), mean absolute relative error (MARE) and Dice coefficient. Successful training was achieved by choosing the correct hyperparameters. The results achieved on the test data set show a high precision of the model, with MSE of 0.0138, Dice coefficient of 0.9055 and prediction accuracy for hornbeam and alder greater than 0.95. Visualization of the results with a heat map and with various graphs confirms the high accuracy of predictions on unseen data. This work contributes to the development of an effective tool for the analysis of ecological data, and with it it is possible to help in the protection of biodiversity and the sustainable management of forest resources.

Keywords: segmentation; classification; *Sentinel-2* satellite; convolutional neural networks; UNET model; mean squared error; mean absolute relative error; dice coefficient; accuracy