

Univerza
v Ljubljani

Fakulteta
*za gradbeništvo
in geodezijo*



MAGISTRSKO DELO

MAGISTRSKI ŠTUDIJSKI PROGRAM DRUGE STOPNJE GEODEZIJA IN GEOINFORMATIKA

Ljubljana, 2022

Univerza
v Ljubljani

*Fakulteta za
gradbeništvo in
geodezijo*



Kandidat/-ka:

Magistrsko delo št.:

Master thesis No.:

Mentor/-ica:

Predsednik komisije:

Somentor/-ica:

Član komisije:

Ljubljana, 2022

POPRAVKI – ERRATA

Stran z napako	Vrstica z napako	Namesto	Naj bo
----------------	------------------	---------	--------

ZAHVALA

Mentorju prof. dr. Krištofu Oštirju ter somentorjema doc. dr. Dejanu Grigillu in mag. Jerneju Nejcu Douganu se zahvaljujem za vse nasvete in pomoč pri nastajanju magistrskega dela.

Še posebej bi se zahvalil mag. Jerneju Nejcu Douganu in kolektivu podjetja Flycom Technologies d.o.o., ki so me sprejeli medse in omogočili izdelavo magistrskega dela, ter hkrati obogatili moje znanje z veliko praktičnega dela.

Posebna zahvala gre vsem domačim, ki so me v celotnem obdobju študija spodbujali, motivirali in podpirali, ter usmerjali skozi življenje. Hvala mami in očetu za vse življenjske modrosti.

Iskrena zahvala tudi vsem prijateljem, sošolcem in cimrom, ki so bili ob meni v dobrih in slabih trenutkih. Preživeli smo veliko skupnega časa in ga obogatili z obilico smeha. Hvala tudi Mojci, ki je življenje v Ljubljani popestrila z dolgimi urami pogovorov na našem balkonu.

BIBLIOGRAFSKO-DOKUMENTACIJSKA STRAN IN IZVLEČEK

UDK:	528.7:556.531/.532(043.3)
Avtor:	Primož Skledar
Mentor:	prof. dr. Krištof Oštir
Somentor:	doc. dr. Dejan Grigillo, mag. Jernej Nejc Dougan
Naslov:	Prepoznavna poplavnih območij iz satelitskih posnetkov Sentinel-2 z modeli konvolucijskih nevronske mreže
Tip dokumenta:	magistrsko delo
Obseg in oprema:	74 str., 18 pregl., 49 sl., 38 en., 56 virov
Ključne besede:	strojno učenje, semantična segmentacija, konvolucijska nevronska mreža, hiperparametri, binarna klasifikacija, MobileNetV2, EfficientNet, Sentinel-2, poplave

Izvleček

Konvolucijske nevronske mreže so v zadnjem desetletju v zelo velikem razvoju in se uporabljajo na skoraj vseh področjih znanosti. Pri nalogah prepoznavne poplavnih območij se v veliki meri uporabljajo nevronske mreže, ki zagotavljajo avtomatizirano prepoznavo le teh z zanesljivimi rezultati. Ti rezultati so pomembni za ocenjevanje škode in pri načrtovanju obnove poplavnega območja. V raziskavi sem uporabil dva modela konvolucijske nevronske mreže, in sicer MobileNetV2 in dve stopnji EfficientNet. Za učenje modela sem uporabil podatke satelita Sentinel-2. Za ločevanje poplavljenega in nepoplavljenega območja sem izdelal lastno zbirko oznak. Izdelali smo program, ki se uporablja za predobdelavo podatkov in učenje modelov. Uporabljene modele sem testiral s spreminjanjem hiperparametrov. Prav tako sem izvedel test spreminjanja po nivoju produkta in izbiro kanalov satelita Sentinel-2. V tretjem delu testiranja sem izboljšal rezultat zgolj z bogatenjem količine podatkov. Po vsakem testiranju sem podatke analiziral in pridobil optimiziran model, kot rezultat, ki je sposoben uspešno prepoznati poplavno območje. V izbranem GIS-u sem uporabil izdelano metodo in jo preizkusil na novih podatkih.

BIBLIOGRAPHIC-DOCUMENTALISTIC INFORMATION AND ABSTRACT

UDC: 528.7:556.531/.532(043.3)
Author: Primož Skledar
Supervisor: prof. dr. Krištof Oštir
Co-supervisor: doc. dr. Dejan Grigillo, mag. Jernej Nejc Dougan
Title: Detection of flooded areas from satellite imagery Sentinel-2 with convolutional neural network models
Document type: Masters Thesis
Notes: 74 p., 18 tab., 49 fig., 38 eq., 56 ref.
Keywords: machine learning, semantic segmentation, convolutional neural network, hyperparameters, binary classification, MobileNetV2, EfficientNet, Sentinel-2, floods

Abstract

Convolutional neural networks have developed a lot over the past decade and are used in almost all fields of science. Neural networks are already used to a large extent for detecting flooded area and provide automated detection with reliable results. These are important for assessing damage and planning the reconstruction of the flood areas. In the study, I used two models of the convolutional neural network MobileNetV2 and two stages of EfficientNet. For learning models, I used Sentinel-2 satellite data. To separate the flooded and not flooded areas, I created my own collection of annotations. We have developed a program that we use to pre-process data and learn models. The models used were tested by changing the hyperparameters. I also performed a product-level change test and a selection of Sentinel-2 satellite channels. In the third part of the tests, I improved the results only by enriching the amount of data. After each test, I analyzed the data and obtained an optimized model as a result that can successfully detect the flooded area. In the selected GMS-GIS, I used the developed method and tested it on new data.

KAZALO VSEBINE

POPRAVKI - ERRATA	I
ZAHVALA	II
BIBLIOGRAFSKO-DOKUMENTACIJSKA STRAN IN IZVLEČEK.....	III
BIBLIOGRAPHIC-DOCUMENTALISTIC INFORMATION AND ABSTRACT	IV
KAZALO VSEBINE	V
KAZALO SLIK	VII
KAZALO PREGLEDNIC.....	X
UPORABLJENE KRATICE IN OKRAJŠAVE	XI
1 UVOD	1
1.1 Namen, cilji, hipoteze in struktura naloge	2
2 TEORETIČNA IZHODIŠČA	4
2.1 Podatki Sentinel-2.....	4
2.1.1 Sentinel-2 L1C	6
2.1.2 Sentinel-2 L2A	7
2.1.3 Problematika oblakov, senc oblakov in snega.....	8
2.1.4 Razpoložljivi podatki.....	9
2.2 Strojno učenje	10
2.3 (Globoke) nevronske mreže.....	11
2.4 Konvolucijske nevronske mreže	14
2.4.1 Uporabnost konvolucijskih nevronske mreže.....	14
2.4.2 Sloji CNN	15
2.4.3 Primeri zgrajenih arhitekturnih modelov CNN	19
2.4.4 Hiperparametri in fine nastavitve	26
2.4.5 Izračun števila parametrov	31
2.4.6 Prekomerno prileganje in tehnike bogatenja količine podatkov.....	32

2.5	Matrike kakovosti naučenih modelov	32
2.5.1	Matrika zamenjav za binarno klasifikacijo	33
2.5.2	Območje pod krivuljo	34
2.5.3	Povprečno razmerje presek:unija - IoU	36
2.6	Metode ločevanja vodnih površin	36
2.6.1	Razširjanje regij	36
2.6.2	Normalizirani diferencialni indeks vode.....	37
3	VIRI IN PREDOBDELAVA PODATKOV	38
3.1	Iskanje dogodkov in priprava geometrije.....	38
3.2	Sentinel Hub in nastavitev slojev	41
3.3	Programski del za prenos podatkov	42
4	ARHITEKTURA ZGRAJENIH MODELOV IN ŠTEVILO PARAMETROV	43
4.1	MobileNetV2	43
4.2	EfficientNet-B0	45
4.3	EfficientNet-B7	48
5	REZULTATI UČENJA MODELOV IN ANALIZA	51
5.1	Rezultat testiranja hiperparametrov po modelu	53
5.2	Rezultat testiranja sprememb podatkovne strukture po modelu	58
5.2.1	Primerjava med tremi modeli CNN	62
5.3	Rezultat testiranja večanja podatkov in števila epoh	63
6	UPORABA MODELA NA NOVIH PODATKIH	66
7	ZAKLJUČEK	69
8	VIRI	71

KAZALO SLIK

Slika 1: L1C razdelitev con UTM na granule (ESA, 2021a)	5
Slika 2: Spektralni kanali Sentinel-2 prostorske ločljivosti 10 m in valovne dolžine kanalov (ESA, 2021a).....	5
Slika 3: Spektralni kanali Sentinel-2 prostorske ločljivosti 20 m in valovne dolžine kanalov (ESA, 2021a).....	6
Slika 4: Spektralni kanali Sentinel-2 prostorske ločljivosti 60 m in valovne dolžine kanalov (ESA, 2021a).....	6
Slika 5: L1C - TOA (levo), L2A - BOA (desno) (ESA, 2021d)	8
Slika 6: Preprosta nevronska mreža (levo), globoka nevronska mreža (desno) (Michael A. Nielsen, 2015)	12
Slika 7: Grafični prikaz enačbe	12
Slika 8: Grafični prikaz aktivacijskih funkcij: a) Sigmoidna, b) usmerjena linearna enota (ReLU), c) Prepustna ReLU, d) Swish.	13
Slika 9: Proces konvolucijske operacije (Tian, 2020)	16
Slika 10: Konvolucija z dodanim robom v vhodnem tenzorju za ohranitev dimenzije v izhodnem tenzorju (Yamashita et al., 2018)	16
Slika 11: Primer konvolucijske operacije v treh korakih, kjer je dimenzija filtra 3 x 3, končni rezultat, viden v koraku c, atributna matrika (ang. feature map) (Yamashita et al., 2018).	17
Slika 12: Enotnost uteži na nevronu.....	18
Slika 13: Procesi podvzorčenja; max - maksimalno podvzorčenje, avg - povprečno podvzorčenje in sto - naključno podvzorčenje (Tian, 2020)	19
Slika 14: Arhitektura U-net-a. Modri štirikotniki predstavljajo matriko značilk z več kanali. Število kanalov je zapisano nad modrimi štirikotniki, medtem ko je dimenzija zapisana na spodnji levi strani le teh. Beli štirikotniki izražajo kopirane matrike značilk. (Ronneberger et al., 2015).....	20
Slika 15: a) blok ostankov, b) blok obrnjenih ostankov (Sandler et al., 2018)	22
Slika 16: Konvolucijski bloki v MobileNetV2. Leva del prikazuje zaporedje konvolucijskih slojev s korakom premikanja filtra 1 piksel. Desna del pa navaja zaporedje virov, s korakom premikanja filtra za 2 piksla. Dwise – globinsko ločljivi konvolucijski sloj, Conv – konvolucijski sloj. Vhod (ang. input) in izhod se na koncu združita (ADD), za kar je potreben konvolucijski sloj. (Dong et al., 2020).....	23

Slika 17: Shematični prikaz osnove arhitekture EfficientNet-B0. MBConv – obrnjena zastojna konvolucija in Conv – konvolucija. (Bazi et al., 2019).....	24
Slika 18: Grafični prikaz razporeditve modelov glede na natančnost in zmogljivost (FLOPS) (Tan & Le, 2019).....	25
Slika 19: Prikaz vhodnih podatkov v normalizacijo primerka po dimenzijah (Karimi, 2021).....	29
Slika 20: Prikaz vhodnih podatkov v paketni normalizaciji po dimenzijah (Karimi, 2021)	30
Slika 21: Prikaz zmanjševanja stopnje učenja po epohah	31
Slika 22: Primer izrisa ROC krivulje in prikaz AUC (Kononenko & Kuhar, 2007).....	35
Slika 23: Primer prikaza pravilne in napovedane vrednosti (Rosebrock, 2022)	36
Slika 24: Sloj topografske karte HERE za prikaz poteka reke	40
Slika 25: Izbrano območje (obarvano rdeče) s pragom 5	40
Slika 26: Izbrano območje (obarvano rdeče) s pragom 20	40
Slika 27: Osnovna sestava bloka - MobileNetV2.....	43
Slika 28: Graf zgrajenega modela MobileNetV2 po blokih	44
Slika 29: Graf bloka "sequential"	45
Slika 30: Osnovna sestava bloka - EfficientNet-B0	46
Slika 31: Graf zgrajenega modela EfficientNet-B0 po blokih.....	47
Slika 32: Graf zgrajenega modela EfficientNet-B7 po blokih.....	49
Slika 33: Grafični prikaz metrike AUC po korakih učenja v prvem delu testiranj - MobileNetV2	54
Slika 34: Potek vrednosti izgube za najboljšo kombinacijo hiperparametrov silvery-field-29 (MobileNetV2)	54
Slika 35: Grafični prikaz metrike AUC po korakih učenja v prvem delu testiranj - EfficientNet-B0 ..	55
Slika 36: Potek vrednosti izgube za najboljšo kombinacijo hiperparametrov honest-water-56 (EfficientNet-B0).....	56
Slika 37: Grafični prikaz metrike AUC po korakih učenja v prvem delu testiranj – EfficientNet-B7 ..	57
Slika 38: Potek vrednosti izgube za najboljšo kombinacijo hiperparametrov sleek-gorge-28 (EfficientNet-B7).....	58
Slika 39: Grafični prikaz metrike AUC po korakih učenja v drugem delu testiranj - MobileNetV2	59
Slika 40: Potek vrednosti izgube pri dveh najboljših testih - MobileNetV2	59

Slika 41: Grafični prikaz metrike AUC po korakih učenja v drugem delu testiranj - EfficientNet-B7	60
Slika 42: Potek vrednosti izgube pri dveh najboljših testih - EfficientNet-B0	60
Slika 43: Grafični prikaz metrike AUC po korakih učenja v drugem delu testiranj - EfficientNet-B7	61
Slika 44: Potek vrednosti izgube pri dveh najboljših testih - EfficientNet-B7	62
Slika 45: Test prekomernega prileganja med tremi modeli	63
Slika 46: Satelitski posnetek (RGB), maska pravih podatkov, maska napovedanih vrednosti, verjetnost napovedanih vrednosti (od levega do desnega posnetka).....	65
Slika 47: Satelitski posnetek v času nepopravljenega območja in prikaz struge v tem času	67
Slika 48: Satelitski posnetek v času poplavljenega območja in prikaz struge v tem času	67
Slika 49: Satelitski posnetek v času poplavljenega območja in prikaz napovedane vrednosti poplavljenega območja.....	68

KAZALO PREGLEDNIC

Preglednica 1: Tipi izdelkov Sentinel-2 (ESA, 2021a)	4
Preglednica 2: Podatkovna struktura produktov L1C in L2A ter njune lastnosti	10
Preglednica 3: Matrika zamenjav za binarno klasifikacijo	33
Preglednica 4: Izbrani dogodki za vektorizacijo podatkov	39
Preglednica 5: Izračun števila parametrov po blokih - MobileNetV2	45
Preglednica 6: Izračun števila parametrov po blokih - EfficientNet-B0	48
Preglednica 7: Izračun števila parametrov po blokih - EfficientNet-B7	50
Preglednica 8: Prikaz števila parametrov in čas učenja na enem koraku (Keras, n.d.)	51
Preglednica 9: Hiperparametri v prvem delu testiranja	52
Preglednica 10: Spremembe podatkov v drugem delu testiranja	52
Preglednica 11: Rezultati prvega dela testiranja - MobileNetV2	53
Preglednica 12: Rezultati prvega dela testiranja – EfficientNet-B0	55
Preglednica 13: Rezultati prvega dela testiranja – EfficientNet-B7	57
Preglednica 14: Rezultati drugega dela testiranja - MobileNetV2	58
Preglednica 15: Rezultati drugega dela testiranja - EfficientNet-B0	60
Preglednica 16: Rezultati drugega dela testiranja - EfficientNet-B7	61
Preglednica 17: Metrike treh najuspešnejših modelov drugega dela testiranja	62
Preglednica 18: Rezultati testiranja večanja podatkov in števila epoh	63

UPORABLJENE KRATICE IN OKRAJŠAVE

NDWI	ang. Normalized Difference Water Index
RF	ang. Random Forest
SVM	ang. Support Vector Machine
CNN	ang. Convolutional Neural Networks
L1C	ang. Level-1C
L2A	ang. Level-2A
TOA	ang. Top of Atmosphere
BOA	ang. Bottom of Atmosphere
VNIR	ang. Visibe/near Infrared
NIR	ang. Near Infrared
SWIR	ang. Short Wave Infrared
MSI	ang. MultiSpectral Instrument
ESA	ang. European Space Agency
AC	ang. Atmospheric Correction
AOT	ang. Aerosol Optical Thickness
WV	ang. Water Vapour
SCL	ang. Scene Classification
CLM	ang. Cloud Mask
CLP	ang. Cloud Probability
CLD	ang. Cloud Probability
SNW	ang. Snow Probablity
ReLU	ang. Rectified Linear Unit
DSC	ang. Depthwise Separable Convolutions
AutoML	ang. Automated Machine Learning
MBConv	ang. mobile inverted bottleneck convolution
RMSProp	ang. Root Mean Square Propagation
BCE	ang. Binary Cross Entropy
DL	ang. Dice loss
LRD	ang. Learning Rate Decay
AUC	ang. Area Under Curve
ROC	ang. Reciever Operating Characteristic
IoU	ang. Intersection over Union

1 UVOD

Opazovanje in napovedovanje naravnih nesreč je bila v preteklosti skoraj nemogoča naloga, vendar se je s pojavom satelitskega opazovanja Zemlje omogočilo njihovo prepoznavanje. Podatki satelitskih posnetkov so trdno povezani s prepoznavo naravnih nesreč, saj predstavljajo enega od glavnih virov informacij o spremembah na zemeljski površini. Za pomembno prelomnico povezave satelitskih podatkov in prepoznavo naravnih nesreč štejemo ustanovitev mednarodnega raziskovalnega programa Vesolje in velike nesreče (ang. The International Charter Space and Major Disasters) leta 2000 (The International Charter Space and Major Disasters, n.d.). Mnogi ponudniki satelitskih posnetkov, kot je Evropska vesoljska agencija, nudijo zraven surovih podatkov tudi aplikacije in storitve za prepoznavo naravnih nesreč. Copernicus, je aplikacija za ravnanje v izrednih razmerah (ang. The Emergency Management Service – Mapping), deluje od leta 2015 in jo financira Evropska komisija (European Union, 2021). Namenjena je učinkovitemu posredovanju satelitskih podatkov pri ukrepanje v naravnih nesrečah. Ponuja širok spekter kartografskih storitev in funkcij za zgodnje opozarjanje.

Od vseh naravnih nesreč so poplave ena najštevilčnejših katastrof, ki vplivajo na človeška življenja, človeško infrastrukturo in svetovno okolje. Najpogostejši vzrok za poplave je povečana količina padavin ali taljenje snega in ledu, kar ima dolgotrajne posledice na poplavno območje. Velikost poplave lahko spremljamo neposredno na terenu ali s pomočjo podatkov daljinskega zaznavanja, ki so pomembni za ocenjevanje škode in posledično za načrtovanja obnove poplavljenega območja. Metode, ki se najpogosteje uporabljajo, za prepoznavanje poplavnega območja iz satelitskih posnetkov so indeksi (normalizirani diferencialni indeks vode) NDWI (ang. Normalized Difference Water Index) ipd.) in tehnika razširjanja regij (ang. region growing). Slabost omenjenih metod je polavtomatski proces, s katerim ne moremo prepoznavati poplav na globalni ravni. Cilj magistrskega dela je popolna avtomatizacija procesa prepoznavne poplav – za kar potrebujemo nove, izpopolnjene, metode procesiranja.

Strojno učenje je pomemben sestavni del rastočega področja podatkovne znanosti. Z uporabo statističnih metod se algoritmi usposabljaajo za klasifikacije ali napovedi. Uveljavljene metode so naključni gozdovi RF (ang. Random Forest) in metoda podpornih vektorjev SVM (ang. Support Vector Machine). Novejši trend strojnega učenja je uporaba globokih nevronske mrež, katere uvrščamo v vejo globokega učenja (ang. deep learning) (W. Li et al., 2016). Preciznega datuma odkritja globokega učenja ne moremo definirati, vendar je Tappert v raziskavi (Tappert, 2019) ugotovil, da Frank Rosenblatt leta 1969 prvi razvije in odkrije vse osnovne lastnosti današnjega globokega učenja. Kot očetje strojnega učenja so bili prepoznani in nagrajeni, s Turingovo nagrado, tudi Yoshua Bengio, Geoffrey Hinton in Yann LeCun. Kasneje se področje močno razvija, kar pripelje, da po letu 2012 nastane več pomembnejših prelomnic, ki spodbudijo revolucijo globokega učenja. Eno od pomembnejših odkritij so konvolucijske nevronske

mreže (Krizhevsky et al., 2017), ki dosegajo dobre rezultate, vendar zahtevajo veliko količino učnih podatkov.

Trend uporabe globokega učenja narašča na vseh znanstvenih področjih. Vedno pogosteje se uporabljajo podatki daljinskega zaznavanja, predvsem satelitski posnetki, iz katerih se izvajajo analize na globalni ravni. Za učenje konvolucijskih nevronske mreže je potrebna velika količina podatkov, ki so v današnjem času na razpolago v velikih količinah. Potreben je tudi proces, ki je največkrat ročen, označevanja podatkov za določitev pravih vrednot na posnetku.

1.1 Namen, cilji, hipoteze in struktura naloge

Namen magistrskega dela je izgradnja modela konvolucijske nevronske mreže za učinkovito prepoznavo poplavnih območij na optičnih satelitskih posnetkih Sentinel-2. Naloga je sestavljena iz več delov z ločenimi cilji. Ti vodijo do zaključnega aplikativnega izdelka. Prvi del zajema sestavo programskega okolja, v katerem poteka učenje. Sledi kombiniranje spektralnih in analitičnih kanalov Sentinel-2 za učenje, izdelava lastne podatkovne baze, glede na zajeto geometrijo z metodo razširjanja regij. Drugi del zajema izdelavo treh konvolucijskih nevronske modelov z različnimi arhitekturami in njihovo testiranje na testnih podatkih. S testi se prepoznajo najboljši hiperparametri za izbrane modele. V drugem delu testiranja različnih kombinacij spektralnih kanalov in analitičnih kanalov (izdelki drugih analiz, kot je maska oblakov ipd.), ki je bil narejen v pričakovanju, da bodo produkti Sentinel-2 L2A zagotovili boljše rezultate. Izmed treh moramo nato izbrati najboljšega in tega na podatkovni strukturi še nadgraditi. Po končani analizi rezultat aplikativno ocenim v uporabnem smislu.

Ker je izbrano področje raziskave zelo široko, se v metodoloških postopkih omejim na izbrane podatke, metode strojnega učenja in hiperparametre za analizo v eksperimentalnem delu.

V nalogi sem postavil tri hipoteze, in sicer:

1. Z izdelanim modelom konvolucijskih nevronske mreže, lahko v zadostni meri, prepoznamo poplavna območja.
2. Produkt satelita Sentinel-2 L2A, torej nivo 2, zagotovi boljše rezultate kot L1C, nivo 1.
3. Izbrani model lahko s hiperparametri izboljšam po metriki AUC z večanjem epoh in tehnikami bogatenja količine podatkov.

V prvem poglavju predstavim kratko zgodovino prepoznavanja naravnih nesreč iz satelitskih podatkov in nastanek globokega učenja, kot veje umetne inteligence, ter namen, hipoteze in cilje magistrskega dela.

Drugo poglavje zajema teoretična izhodišča uporabljenih podatkov in metod dela. Predstavljena je podatkovna struktura podatkov Sentinel-2. Opredeljena je razlika med podatki Sentinel-2 nivoja L1C in L2A, ter kako se opravi prehod na L2A. Nakar je izpostavljena problematika oblakov, senc oblakov in

snega. Predstavljeni so razpoložljivi podatki, ki so dostopni na platformi Sentinel Hub. Za izdelavo oznak na učnih podatkih sta opredeljeni dve metodi, in sicer metoda normalizirani diferencialni indeks vode in metoda razširjanje regij, ki se uporabi v eksperimentalnem delu. V osnovi je predstavljeno področje strojnega učenja in globokih nevronske mreže. Podrobneje je razložena struktura konvolucijske nevronske mreže (Convolutional Neural Networks - CNN) in opisani sta osnovni arhitekture dveh modelov. Razlaga hiperparametrov je omejena na te, ki so uporabljeni v eksperimentalnem delu. Prav tako je dodana definicija izračuna števila parametrov modela, ki v osnovi predstavlja uteži, naučene med procesom učenja, in je metrika uteži, ki prispeva k napovedi moči modela, prekomerno prileganje modela učnim podatkom in kakšne so potencialne posledice. Ker sem moral vsak model ovrednotiti, drugo zaključujem z krajšo obravnavo metrik kakovosti naučenih modelov.

Proces strojnega učenja sestavljajo pridobivanje in urejanje podatkov, izgradnja modelov, treniranje modelov, evalvacija rezultatov ter uporaba le-teh. Tako so sestavljena naslednja tri poglavja. Tretje poglavje, pridobivanje podatkov, zajema iskanje dogodkov ter pripravo geometrije, na platformi Sentinel Hub in delovanje programskega dela za pridobivanje podatkov. Uporabljeni so trije modeli, zato je v četrtem poglavju grafično in opisno opredeljena arhitektura modelov in izračun števila parametrov vsakega modela. Končni del procesa strojnega učenja je evalvacija rezultatov, ki je tabelarično, grafično in opisno, v treh delih, analizirana v petem poglavju. Prvi del zajema analizo testiranja hiperparametrov po modelu. Drugi del obravnava testiranje sprememb podatkov po modelu. Tretji del predstavi test večanja podatkov in števila epoh učenja.

Po končanem eksperimentalnem delu in pridobljenem optimalnem modelu in njegovi predstavitvi rezultatov v šestem poglavju aplikativno prikažem uporabo le tega na novih bazah podatkov. V zaključnem poglavju predstavim glavne ugotovitve, evalvacijo hipotez in rezultate ciljev.

2 TEORETIČNA IZHODIŠČA

2.1 Podatki Sentinel-2

Copernicusova evropska misija Sentinel-2 je namenjena za pridobivanje širokopasovnih, visoko ločljivih in multispektralnih posnetkov zemeljske površine. Misijo sestavljata dva satelita, ki sinhrono krožita v orbiti v stalni razdalji in medsebojni legi 180° . Monitoring zemeljske površine poteka v pasu širine 290 km, celotno Zemljo sistem optično posname v 10 dneh, v primeru obeh satelitov v 5 dneh.

Podatki optičnih posnetkov vsebujejo 13 spektralnih kanalov, in sicer od vidnega do bližnjega in kratkovalovnega infrardečega spektra elektromagnetnega valovanja. Ker vsebujejo širok nabor podatkov, se uporabljajo v številnih aplikacijah, kot so, pomoč varnostnim službam, odzivom v sili, spremljanjem zemljišč in drugod (ESA, 2021b). Monitoring zemeljske površine je s satelitskimi posnetki Sentinel-2A postal operativen leta 2015, zato se uporablja tudi za spremljanje sprememb vegetacije in vodnih površin.

ESA ponuja dva izdelka misije Sentinel-2:

- Nivo-1C (ang. Level-1C, L1C)
- Nivo-2A (ang. Level-2A, L2A)

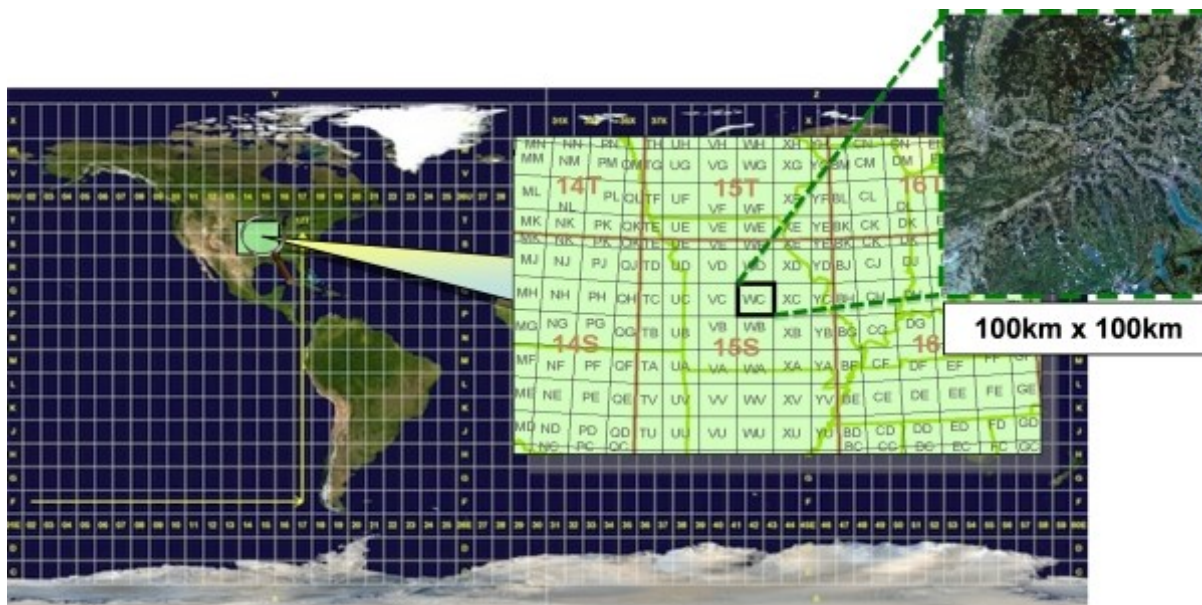
Prav tako obstajajo izdelki nivo-0, nivo-1A in nivo-1B, vendar niso obče dostopni uporabnikom.

Preglednica 1: Tipi izdelkov Sentinel-2 (ESA, 2021a)

Ime izdelka	Opis	Produkcija in distribucija	Velikost podatkov
L1C	Odboj na vrhu atmosfere (TOA)	Sistematično generiranje in sprotne distribucije	600 MB (vsak 100 x 100 km ²)
L2A	Odboj na dnu atmosfere (BOA)	Sistematično generiranje in sprotne distribucije ali generiranje na strani uporabnika (z uporabo orodja Sen2Cor)	800 MB (vsak 100 x 100 km ²)

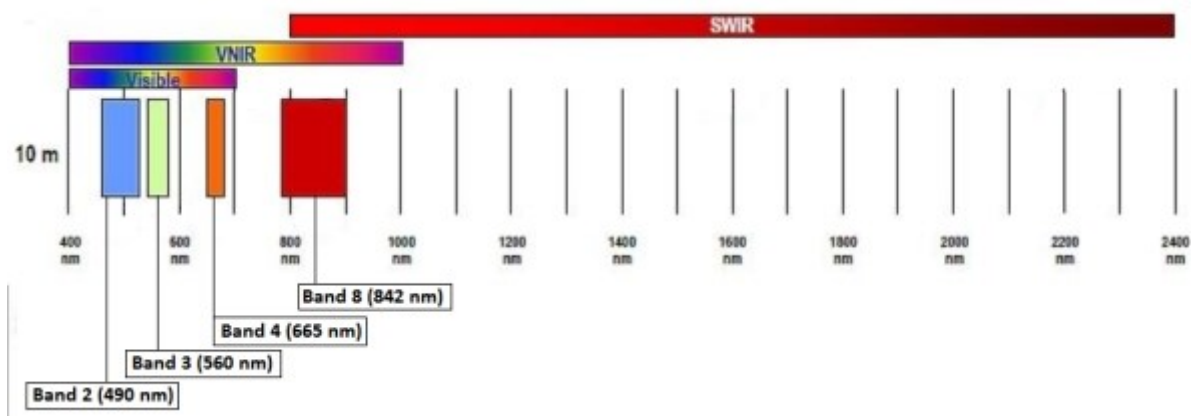
Posamezni izdelek je sestavljen iz granul ali plošč (ang. tiles), ki imajo stalno velikost znotraj ene orbite. Za L1C in L2A so granule tako imenovani orto-posnetki dimenzije 100 x 100 km² v kartografski projekciji UTM / WGS84. Pri tem je pomembno izpostaviti UTM, saj je osnova za granule, ker deli opazovano zemeljsko površje na 60 con (Slika 1).

Vsaka cona ima dimenzijo 6° geografske dolžine in 8° geografske širine in sestavlja več granul. Podatki jih pokrivajo v celoti ali delno, kar je odvisno od prekrivanja granul z razdelitveno mrežo UTM con (ESA, 2021c).



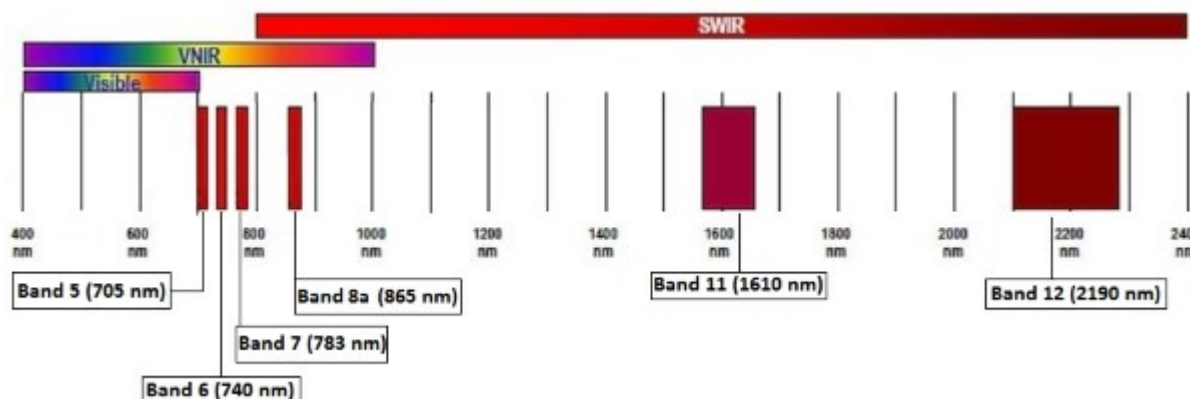
Slika 1: LIC razdelitev con UTM na granule (ESA, 2021a)

Za opredelitev ločljivosti, ki opredeljuje zmogljivost sistema je potrebno biti pozoren na časovno, prostorsko in radiometrično ločljivost. Kot sem omenil, časovna ločljivost optično posname v 5 dneh z dvema satelitoma. Prostorska ločljivost predstavlja velikost najmanjših objektov na opazovanem posnetku. Zaradi različnih senzorjev se spektralni kanali, po prostorski ločljivosti, delijo v tri skupine po 10 m, 20 m in 60 m. Radiometrična ločljivost instrumenta pomeni sposobnost, da razlikuje med odbojnostjo žarkov, kar se zapisuje v enoti nanometer (nm). Večja kot je radiometrična ločljivost senzorja, manjše razlike odbite intenzitete lahko loči. Na podoben način ločimo spektralno ločljivost, in sicer na vidno (VNIR), na bližnjo infrardečo (NIR) ter kratkovalovno infrardečo (SWIR). Slika 2 prikazuje prvo skupino spektralnih kanalov (B2, B3, B4 in B8) prostorske ločljivosti 10 m. Ob vsakem kanalu je dopisana valovna dolžina v nanometrih.



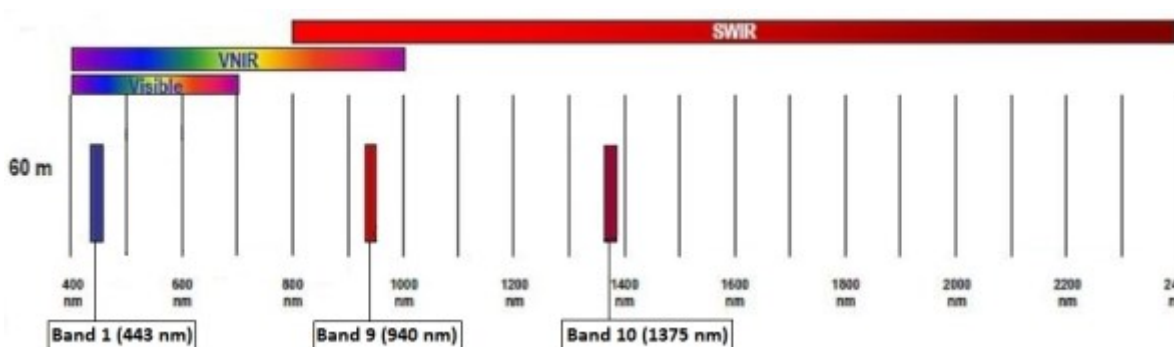
Slika 2: Spektralni kanali Sentinel-2 prostorske ločljivosti 10 m in valovne dolžine kanalov (ESA, 2021a)

Druga skupina kanalov ima prostorsko ločljivost 20 m. Prvi štirje (B5, B6, B7 in B8A) kanali so v VNIR za vegetacijski rob ter kanal B11 in B12 za prepoznavo snega, oblakov, ledu ali za oceno vlage v vegetaciji (Slika 3).



Slika 3: Spektralni kanali Sentinel-2 prostorske ločljivosti 20 m in valovne dolžine kanalov (ESA, 2021a).

V zadnji skupini so trije spektralni kanali (B1, B9 in B10) prostorske ločljivosti 60 m, ki so namenjeni za popravek vpliva atmosfere in določanje prisotnosti oblakov (Slika 4).



Slika 4: Spektralni kanali Sentinel-2 prostorske ločljivosti 60 m in valovne dolžine kanalov (ESA, 2021a)

2.1.1 Sentinel-2 L1C

Za zajem podatkov se kot senzor uporablja multispektralni instrument MSI (ang. MultiSpectral Instrument), ki pridobiva 13 spektralnih kanalov. L1C vključuje tako geometrične in radiometrične popravke, kot tudi orto-rectifikacijo in prostorsko registracijo na globalni referenčni sistem z natančnostjo piksla. V (ESA, 2021c) procesiranje L1C definirajo v korakih:

- združenje granul,
- ponovno vzorčenje mreže (orto-rectifikacija),
- ponovno vzorčenje vsakega spektralnega kanala,

- izračun odbojnosti TOA,
- izračun mask oblakov in kopno/voda,
- kompresija posnetkov.

Svetloba se v vrhnjem delu atmosfere odbija med drugim tudi od plasti aerosola in vodne pare. Za naloge, kjer je ustrezna vrednost spektralnega kanala bistvenega pomena, je potreben atmosferski popravek, na primer s procesorjem Sen2Cor. O tem podrobneje v naslednjem poglavju.

Značilna uporaba tega izdelka je kartiranje Zemeljskega površja, spremljanje sprememb tal za ravnanje v izrednih razmerah, spremljanje vegetacije, spremljanje požganih površin ipd.

2.1.2 Sentinel-2 L2A

Vhodni podatki za izdelavo izdelka L2A so izdelki L1C. Od leta 2018 ta zagotavlja satelitske posnetke z odbojem z dna atmosfere (ang. Bottom of Atmosphere, BOA). Uporabnik lahko postopek izdelave opravi individualno z uporabo orodja Sen2Cor, ki se nahaja v brezplačnem odprtokodnem orodju Sentinel-2 (ang. Sentinel-2 toolbox). Orodja so dostopna v programu SNAP, v katerem so združene vse Sentinel Toolbox knjižnice. ESA ga uporablja za opazovanje satelitskih posnetkov in mnoge analize, ki jih vsebuje. Za podporo lastnim misijam razvija razna orodja, ki so ravno tako sestavni del programa SNAP (European Space Agency, 2021).

Procesor Sen2Cor je razvil Telespazio VEGA Deutschland GmbH na zahtevo ESA in nudi več izdelkov (Main-Knorn et al., 2017). Glavni namen procesorja je atmosferski popravek (ang. Atmospheric Correction, AC) produkta L1C, s katerim se kot rezultat dobi tudi optično debelino aerosola (ang. Aerosol Optical Thickness, AOT), vodno paro (ang. Water Vapour, WV), klasifikacijo (ang. Scene Classification, SCL) in kazalnike za verjetnost oblakov in snega.

Atmosferski popravki se izvajajo z uporabo niza tabel za iskanje. Več tabel omogoča popravek v zimskem ali poletnem obdobju na srednji geografski širini, vsebnostjo aerosola in ozona na podežlju ter morja glede na geografsko lego (Main-Knorn et al., 2017).

Sen2Cor je uporaben še za druge namene, ki so omenjeni v (Main-Knorn et al., 2017). Čeprav je bil razvit za aplikacije na kopnem, ga je mogoče uporabiti tudi na vodnih površinah, vendar z uporabo AOT ocenjenega na območju kopnega. Procesor pri tem ne upošteva učinkov vodne površine, kot je sončni odsev.

Vpričo naštetih izdelkov so izhodni podatki tudi spektralni kanali z atmosferskim popravkom in ohranjajo enako prostorsko ločljivost kot izdelki L1C. Izjema je le kanal B10, saj nima podatka o zemeljski površni, ampak o oblakih cirus, zato ga v izdelku L2A ni.

Slika 5 prikazuje razliko med L1C in L2A produktom.



Slika 5: L1C - TOA (levo), L2A - BOA (desno) (ESA, 2021d)

2.1.3 Problematika oblakov, senc oblakov in snega

Pomemben korak pred obdelavo satelitskih posnetkov je zaznavanje oblakov in upoštevanje le-teh v obdelavi.

Dva znana izdelka sta maska oblakov (ang. Cloud Mask, CLM) in verjetnost oblakov (ang. Cloud Probability, CLP). Izdelka sta rezultat algoritma s2cloudless, ki temelji na principu strojnega učenja. Izdelalo ga je podjetje Sinergise d.o.o., ki je sodelovalo tudi na CMIX¹.

Podatkovna struktura:

- CLM: tri možne vrednosti → 0 (ni oblakov), 1 (oblaki), 255 (ni podatka)
- CLP: interval verjetnosti → 0 - 255

Izračun je bil narejen na podlagi spektralnega kanala B10 in je na voljo v prostorski ločljivosti 160 m (Repse et al., 2020).

¹ CMIX – Cloud Masking Inter-comparison Exercise je mednarodna skupna pobuda za medsebojno primerjavo algoritmov za zaznavanje oblakov iz posnetkov posnetih z visoko-ločljivimi optičnimi senzorji.

Poznamo tudi izdelka verjetnosti oblakov (ang. Cloud Probability, CLD) in verjetnosti snežnih površin (ang. Snow Probability, SNW), ki sta izdelana z orodjem Sen2Cor². Podatki so v obliki verjetnosti v intervalu 0–100 in v prostorski resoluciji 60 m ali 20 m. Izdelana sta z uporabo algoritma SCL, ki uporabi lastnosti svetlobne odbojnosti območja, da ugotovi prisotnost oblakov in snega.

V eksperimentalnem delu sem uporabil sloje CLM, CLP in CLD, ki so dostopni na SentinelHub.

2.1.4 Razpoložljivi podatki

V preglednici 2 je navedena podatkovna struktura za učenje modelov strojnega učenja in tematike poplav razpoložljivih na SentinelHub. Dodatno je navedeno, v kakšni ločljivosti so dostopni podatki ter kakšne intervalne vrednosti imajo kanali. Podobno kot pri nekaterih drugih uporabljenih algoritmihi, je tudi pri strojnem učenju pomembno, da imamo podatke normalizirane na enake intervale med 0 in 1 ter poenoten format zapisa podatkov. Zato so navedeni originalni intervali vrednosti. Uporabil sem format zapisa podatkov float³, saj se z njim pridobi večja barvna globina.

² Sen2Cor konfiguracija in uporabniška navodila (ESA, 2018)

³ Float – decimalno število

Preglednica 2: Podatkovna struktura produktov L1C in L2A ter njune lastnosti

Ime	Opis	Prostorska ločljivost [m]	Interval vrednosti	S2-L1C	S2-L2A
B01	Ultra modra – obalni aerosol	60	0–4095	DA	DA
B02	Modra	10	0–4095	DA	DA
B03	Zelena	10	0–4095	DA	DA
B04	Rdeča	10	0–4095	DA	DA
B05	VNIR ¹ – Rdeči rob vegetacije	20	0–4095	DA	DA
B06	VNIR ¹ – Rdeči rob vegetacije	20	0–4095	DA	DA
B07	VNIR ¹ – Rdeči rob vegetacije	20	0–4095	DA	DA
B08	NIR ²	10	0–4095	DA	DA
B8A	Ozka NIR ²	20	0–4095	DA	DA
B09	SWIR ³ – vodna para (oblaki)	60	0–4095	DA	DA
B10	SWIR ³ – cirrus	60	0–4095	DA	NE
B11	SWIR ³	20	0–4095	DA	DA
B12	SWIR ³	20	0–4095	DA	DA
SNW	Verjetnost snega	20	0–100	NE	DA
CLD	Verjetnost oblakov	20	0–100	NE	DA
CLP	Verjetnost oblakov	160	0–255	DA	DA
CLM	Maksa oblakov	160	0, 1, 255	DA	DA
dataMask	Maksa podatki / brez podatkov	N/A	0, 1	DA	DA

¹ Vidna/bližnja infrardeča (ang. Visibe/near Infrared, VNIR)

² Bližnja infrardeča (ang. Near Infrared, NIR)

³ Kratkovalovna infrardeča (ang. Short Wave Infrared, SWIR)

2.2 Strojno učenje

Alana Turinga si je kot začetnik teoretičnega računalništva in umetne inteligence zastavil vprašanje: Ali lahko stroji razmišljajo? Po tem je znan tudi Turingov test, ki ga je definiral leta 1950. Jedro ideje je, da ne moremo ločiti, ali gre za človeka ali stroj. Na eni strani je izpraševalec, na drugi pa računalnik ali

človek. Pogovor poteka preko ekrana in tipkovnice. V kolikor izpraševalec ne more ločiti človeka od stroja, potem se obravnava, da je sistem inteligenten. Test je subjektiven, zato je kot tak podvržen kritikam.

Strojno učenje je veja umetne inteligence in računalništva, ki omogoča samodejno izboljševanje algoritmov na podlagi učnih podatkov. To se lahko obravnava tudi kot preslikava, kjer se preslikajo vhodni podatki v izhodne s funkcijo, ki se ustvari z učenim procesom. Točne funkcije ni mogoče nikoli ustvariti, lahko se pridobi najboljši približek želene funkcije.

Strojno učenje lahko razdelimo v tri skupine (Varone et al., 2020):

- nadzorovano učenje,
- nenadzorovano učenje in
- spodbujevalno učenje.

Pri nadzorovanem učenju operiramo z vhodnimi podatki, zraven katerih so potrebne maske pravih vrednosti. Določi se arhitektura modela, ki preslika vhodne podatke v ciljne in shranimo prave vrednosti. Primer je učenje iz ročne pisave, kjer na vsaki posamezni sliki definiramo, kaj predstavljajo črke.

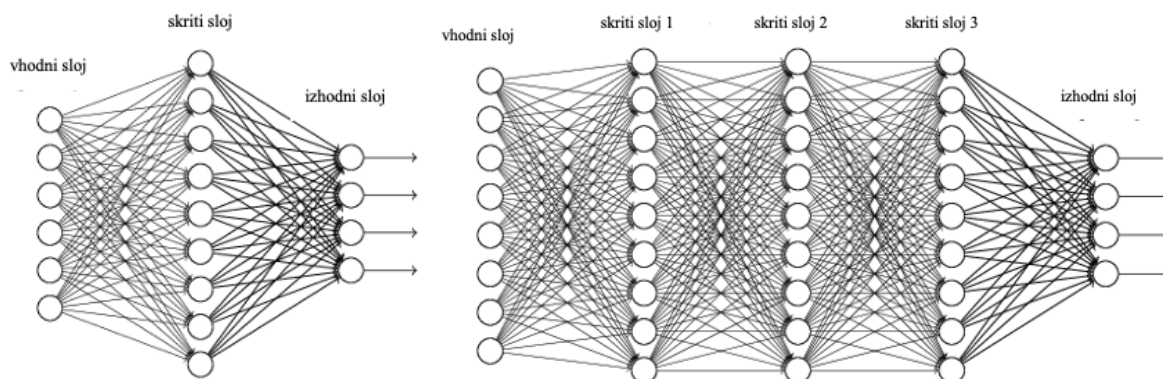
V nadaljevanju bomo uporabljali nadzorovano učenje.

Tipični algoritem strojnega učenja je sestavljen iz treh komponent (Tamir, 2020):

1. postopka odločanja: preračuni vhodnih podatkov in vrnitev približkov na vrsto vzorca v podatkih, ki jih išče zastavljen algoritem,
2. funkcije napake: izračuna se glede na razliko približkov iz prejšnjega koraka in znanih pravih vrednosti (če so na voljo) in
3. postopka optimizacije: algoritem pogleda, kje je naredil napako in posodobi odločanje za zmanjševanje napake.

2.3 (Globoke) nevronske mreže

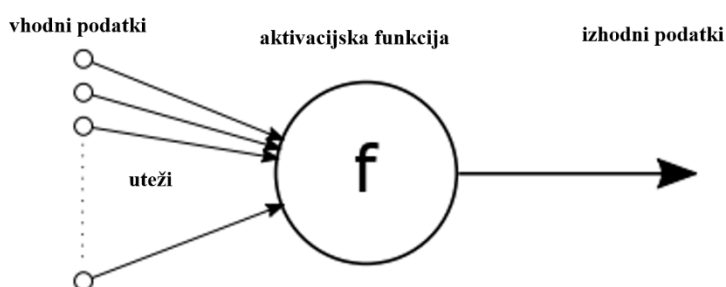
Nevronske mreže so računalniški sistemi, katerih glavni gradnik je posamezen nevron. Ta ima vlogo matematične funkcije, ki na podlagi vhodnih podatkov izračuna izhodno vrednost in običajno predstavlja kar rezultat nelinearne vsote vhodnih vrednosti (Keijsers, 2010). Najpreprostejša nevronska mreža (Slika 6, levo) je sestavljena iz treh slojev: vhodni, skriti in izhodni sloj. Ko se preide do kompleksnejših mrež, je več skritih slojev, ki se imenujejo globoke nevronske mreže (Slika 6, desno).



Slika 6: Preprosta nevronska mreža (levo), globoka nevronska mreža (desno) (Michael A. Nielsen, 2015)

Pri izračunu rezultatov se upoštevajo uteži w , pristranska vrednost b in aktivacijska funkcija A , da se iz vhodnih podatkov x dobi rezultat y (Slika 7):

$$y = A(w^T x + b) \quad (1)$$



Slika 7: Grafični prikaz enačbe

Aktivacijska funkcija opravi glavno vlogo pri učenju nevronske mreže, saj bi brez nje sledila linearna regresija. Omogoča obravnavati zapletene, nelinearne podatke, s katerimi se gradijo kompleksne arhitekture modelov.

Izbira aktivacijske funkcije je odvisna od naloge ob pogoju, da je odvedljiva. Spodaj so naštet štiri pogosto uporabljene funkcije, s poudarkom, da se trenutno skoraj izključno uporablja samo še usmerjena linearna enota (ReLU), oziroma prepustna usmerjena linearna enota (Leaky ReLU). V nadaljevanju so pripisane in grafično predstavljene tudi enačbe (Slika 8).

- Sigmoidna funkcija (ang. sigmoid function):

$$f(x) = \sigma(x) = \frac{1}{1+e^{-x}} \quad (2)$$

Ima izhodne vrednosti med 0 in 1, zato se pogosto uporablja za klasifikacijo v dva razreda. Ni primerna za skrite sloje globokih nevronske mrež, ker odvodi hitro postanejo premajhni, kar upočasni učenje.

- Usmerjena linearna enota (ang. Rectified Linear Unit, ReLU):

$$f(x) = \max(0, x) \quad (3)$$

Ima izhodne vrednosti med 0 in ∞ . Uporablja se v skritih slojih globokih nevronske mreže.

- Prepustna usmerjena linearna enota (ang. Leaky ReLU):

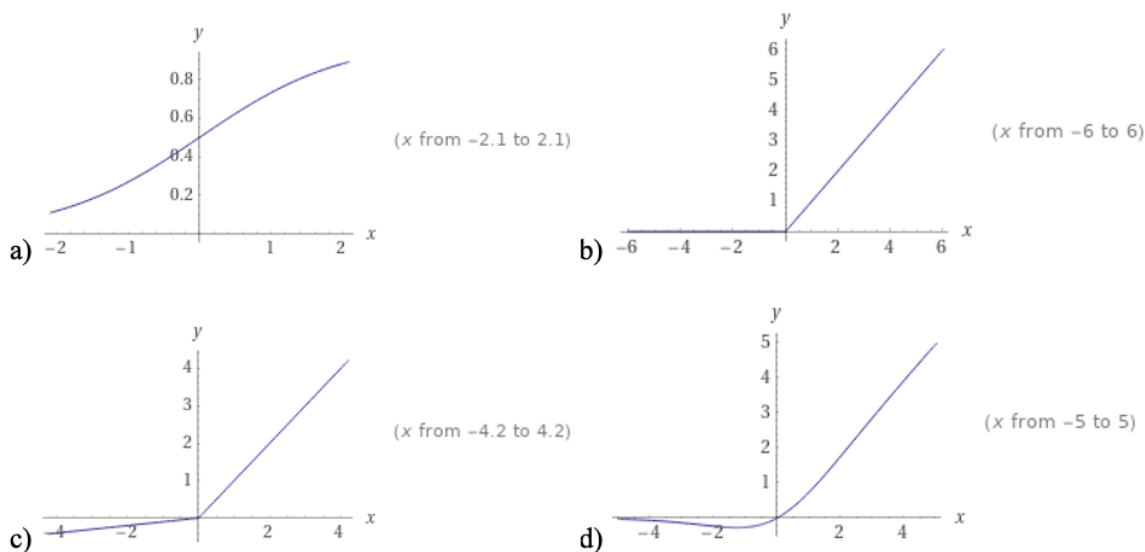
$$f(x) = \max(0.01x, x) \quad (4)$$

Spremeni funkcijo, da dovoli majhne negativne vrednosti, ko je vhod manjši od nič. Uporablja se predvsem za notranje sloje globokih nevronske mreže.

- Swish:

$$f(x) = x \times \sigma(x) = x \times \left(\frac{1}{1+e^{-x}} \right) \quad (5)$$

Ena od novejših aktivacijskih funkcij, ki so jo razvili Ramachandran et al. (Ramachandran et al., 2018) Uporabljena je tudi v modelu EfficientNet, ki ga uporabimo v eksperimentalnem delu. Funkcija je neomejena navzgor in navzdol. Prav tako je gladka in ne monotona, po čemer se razlikuje od večine drugih aktivacijskih funkcij.



Slika 8: Grafični prikaz aktivacijskih funkcij: a) Sigmoidna, b) usmerjena linearna enota (ReLU), c) Prepustna ReLU, d) Swish.

Po prvem rezultatu, ob poznani ciljni vrednosti, se model oceni tako, da se izračuna napaka. Namen je, da se jo minimizira, kar se doseže z algoritmom vzratnega razširjanja napake (ang. backpropagation). Izračuna se napaka med izhodno in ciljno vrednostjo, z razširitvijo do vhodne vrednosti. Na primeru preproste nevronske mreže z enim skritim slojem se izračunajo parcialne odvode funkcije glede na uteži med izhodnim in skritim slojem, nato se postopek ponovi med skritim in vhodnim slojem. Enak

postopek bi bil za globoke nevronske mreže, kjer bi se izračun naredil med vsemi skritimi sloji. Cilj algoritma je iskanje minimuma funkcije napake z gradientnim spustom (Amrutha & Remya Ajai, 2018).

Globoke nevronske mreže se lahko delijo tudi po kompleksnosti, ki se računajo s številom parametrov. Izračun se naredi za vsak posamezen sloj in predstavljajo število elementov v procesu učenja. Predstavljajo uteži, ki jih model spreminja z vzvratnim razširjanjem napake.

2.4 Konvolucijske nevronske mreže

Osnovni gradnik je nevron, ki vsebuje učinkovite uteži in pristranskost. Vsak nevron dobi vhodne podatke, aplicira aktivacijsko funkcijo in vrne izhodno vrednost upoštevajoč vektorski izdelek. Kot rezultat vrne napoved, torej verjetnosti napovedanih razredov ali regresijsko napoved ob uporabi kriterijske funkcije (ang. loss function).

Konvolucijska nevronska mreža (ang. Convolutional Neural Networks – CNN) je slična preprosti nevronske mreži, ki je sestavljena iz več slojev. Posamezni sloj sprejme večdimenzionalni tenzor, ki ga z odvedljivo funkcijo preslika v nov večdimenzionalni tenzor. Primer na satelitskih posnetkih je dimenzija 3D (širina, višina, globina).

Splošno definirane so konvolucijske nevronske mreže, ki so sestavljene iz več računskih blokov, kot so konvolucijski, združevalni in standardni polno povezani izhodni sloji, oziroma nivoji. Zasnovane so za samostojno in prilagodljivo učenje z algoritmi vzvratnega razširjanja napake.

CNN je poseben tip globokih nevronske mrež, ki se uporablja za številne naloge, med katerimi je še izrazito osredotočenje v podatke z mrežno strukturo, kot so slike. Zasluge za to imata LeCun in Bengio (LeCun et al., 1998), ki sta želela replicirati mačkin korteks (vidno zaznavanje) z uporabo posebne arhitekture nevronske mrež. Tako sta razvila model LeNet, ki je bil sprva namenjen klasifikaciji ročno napisanih črk. CNN sta matematično zasnovala iz treh tipov slojev: konvolucijski, združevalni in polno povezani. Prva dva sta namenjena za izvleček značilk oz. lastnosti podatkov, medtem ko polno povezovalni sloj izlušči bistvene lastnosti in vrne končen rezultat (pri klasifikaciji so to razredi).

2.4.1 Uporabnost konvolucijskih nevronske mrež

Konvolucijske nevronske mreže se uporabljajo na številnih področjih za različne namene. Z njimi se srečujemo v vsakdanjem življenju in rešujemo probleme, kot so klasifikacija, detekcija objektov in semantična segmentacija.

Pogosta uporaba klasifikacije se vrši na 2D podatkih, torej slikovna klasifikacija. Temeljno je, da model prepozna objekt na sliki in mu dodeli razred (npr. pes, ogledalo ipd.) ali verjetnost, da pripada določenemu razredu.

Naslednja stopnja je detekcija objektov, ki jim najprej dodeli razrede, kot pri klasifikaciji, nato jih še na sliki locira. Objekt se locira tako, da nanj izriše okvir, znotraj katerega se nahaja klasificiran objekt. Objektov je na sliki lahko več. Ta metoda detekcije se uporablja pri video nadzorih, prepoznavi iz slik v medicini, navigacijo robotov in mnoge druge.

Zadnja izmed naštetih problemov je semantična segmentacija, ki opravi klasifikacijo za vsak piksel posebej. Danes je to eden ključnih izzivov računalniškega vida, saj stremi k popolnemu razumevanju okolice in scene. Pomembnost razumevanja okolice in scene, kot računalniškega problema, poudarja dejstvo, da vedno več aplikacij uporablja prepoznavo iz slik in se mnogokrat tudi nanj zanaša oziroma je vedno večje zaupanje v rezultate strojnega učenja. Določene aplikacije vključujejo avtonomno vožnjo, virtualno resničnost, interakcijo med človekom in računalnikom ipd. Z vedno večjo uporabo globokega učenja se v zadnjih letih mnogi semantični problemi spopadajo z globokimi arhitekturami, najpogosteje konvolucijskimi nevronskimi mrežami, ki lahko zagotavljajo visoko točnost in učinkovitost.

V eksperimentalnem delu se ukvarjam s semantično segmentacijo, kjer se za vsak piksel določa ali gre za poplavo (binarni problem).

2.4.2 Sloji CNN

2.4.2.1 Konvolucijski sloj

Konvolucija je poseben tip linearnih operacij, ki se ukvarja s pridobivanjem značilnosti iz posnetkov. Pri tem se matrika, ki se imenuje jedro ali filter (ang. kernel), uporablja za preračunavanje vseh elementov vhodne matrike, tako imenovani vhodni tenzor (ang. input tensor). Filter ima majhno dimenzijo (širina, višina), manjšo od vhodnih podatkov, le globina mora biti enaka. Pomikamo ga na vhodnih podatkih po širini in višini ter na vsaki poziciji izračunamo skalarno vrednost vhoda in filtra. Po opravljenem izračunu celotnih vhodnih podatkov, se kot rezultat dobi aktivacijska matrika, oziroma matrika značilnosti (ang. activation map, feature map). Na slikah 9 in 10 sta prikazana vhodna matrika in filter, s katerim izračunamo rezultat. Na sliki 9 je filter zapisan v obliki enačbe, na sliki 10 pa v obliki matrike značilnosti (ang. feature map). Ta postopek se ponavlja, dokler ne obdelava vseh vhodnih podatkov po vsakem pikslu posebej. Konvolucijo matematično opišemo z enačbo:

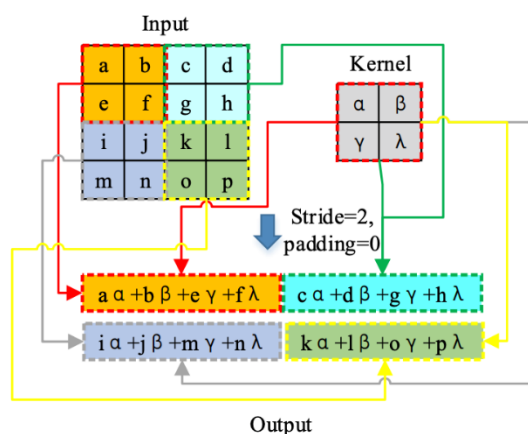
$$(I \times K)(i, j) = \sum_m \sum_n I(m, n) K(i - m, j - n), \quad (6)$$

kjer »I« predstavlja vhodno matriko in »K« filter.

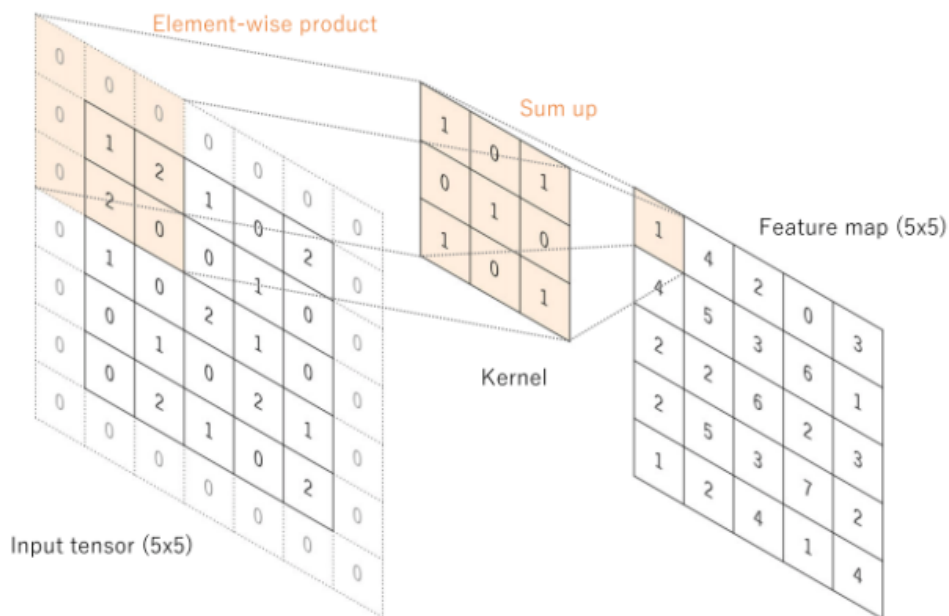
Smisel uporabe različnih filtrov je, da se identificirajo različne lastnosti vhodne matrike. Trije hiperparametri, ki definirajo konvolucijsko operacijo, so: dimenzija filtra, število filtrov in korak premikanja filtra (ang. stride). Običajno, v spodnjem primeru na sliki 10, je to dimenzija 3 x 3. Uporabljata se tudi dimenziji 5 x 5 in 7 x 7. Drugi hiperparameter, število filtrov, definiramo s številom

razredov, ki bodo v matriki značilnik oz. končnem rezultatu. V eksperimentalnem delu se zazna binarni problem, torej je/ni poplava.

Tretji hiperparameter, korak premika, pove, na kolikih pikslih se bo pomikal filter na vhodni matriki. Slika 9 prikazuje 2 koraka in slika 10 1 korak. Velikost koraka zelo vpliva na čas učenja.



Slika 9: Proces konvolucijske operacije (Tian, 2020)



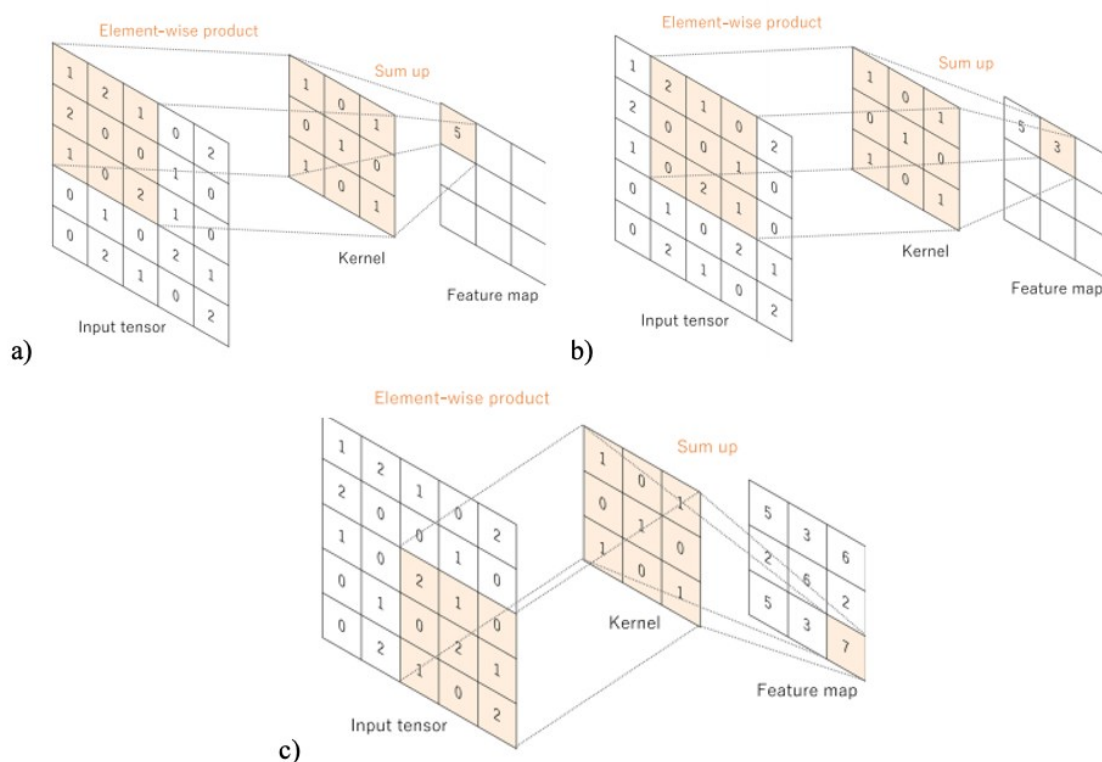
Slika 10: Konvolucija z dodanim robom v vhodnem tenzorju za ohranitev dimenzije v izhodnem tenzorju (Yamashita et al., 2018)

Na primeru s slike 11 je razvidno, da se center filtra ne dotakne roba vhodne matrike, kar povzroči, da je matrika značilnik manjše dimenzije. Problem se reši z dodajanjem obrobe (ang. padding) na rob vhodne

matrike, ki je običajno sestavljena iz ničel. Primer dodajanja obrobe je prikazan na sliki 10, ki ima dodan rob, kar pomeni, da je obroba enaka 1. Tako je omogočeno centru filtra, da doseže vse robove vhodne matrike. Upoštevati se mora, da podatki na robovih izhodnega tenzorja ne bodo izražali točne napovedi. Ob vsem omenjenem se lahko izračuna velikost izhodne matrike po enačbi:

$$O = \frac{W-F+2P}{S}, \quad (7)$$

kjer »W« predstavlja velikost vodne matrike, »F« velikost filtra, »P« velikost obrobe, »S« velikost koraka pomika in »O« velikost izhodne matrike.



Slika 11: Primer konvolucijske operacije v treh korakih, kjer je dimenzija filtra 3 x 3, končni rezultat, viden v koraku c, atributna matrika (ang. feature map) (Yamashita et al., 2018).

Za konvolucijski sloj sta prav tako značilni lokalna povezljivost in enakost uteži. Zaradi teh dveh lastnosti je CNN pri učenju veliko hitrejši v času procesiranja od ostalih globokih nevronske mrež.

- Lokalna povezljivost:

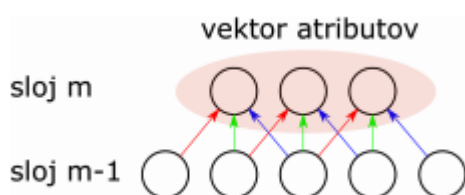
Na sliki 11 lokalni vzorec predstavlja obarvani del vhodne matrike. Posamezen nevron na izhodnem sloju je le lokalno povezan z lokalno definirano podmnožico nevronov v predhodnem sloju. S je zagotovljeno, da je izhodni nevron posredno odvisen le od lokalnih vzorcev vhodnega podatka.

Lokalno povezljivost omogočajo filtri. Ker je dimenzija filtra manjša od dimenzije vhodnih podatkov, so vsi nevroni v matriki značilno povezani le z lokalnim delom vhodnega tenzorja. Prednost lokalne

povezljivosti je naučljivost filtrov, ki maksimalno ustrezajo lokalnim delom vhodnega tenzorja, da izkoristi korelacijo med sosednjimi nevroni.

- Enakost uteži:

Vsakemu lokalnemu vzorcu je dodeljen filter ali skupek uteži. Ker s filtrom prehaja čez celotno območje, pomeni, da so uteži enotne na celotnem območju, kar velja tudi v enostavnem enodimenzionalnem primeru, kot je prikazano na sliki 12. Z barvami je nazorno prikazana enotnost uteži na vsakem nevronu v sloju m .



Slika 12: Enotnost uteži na nevronu

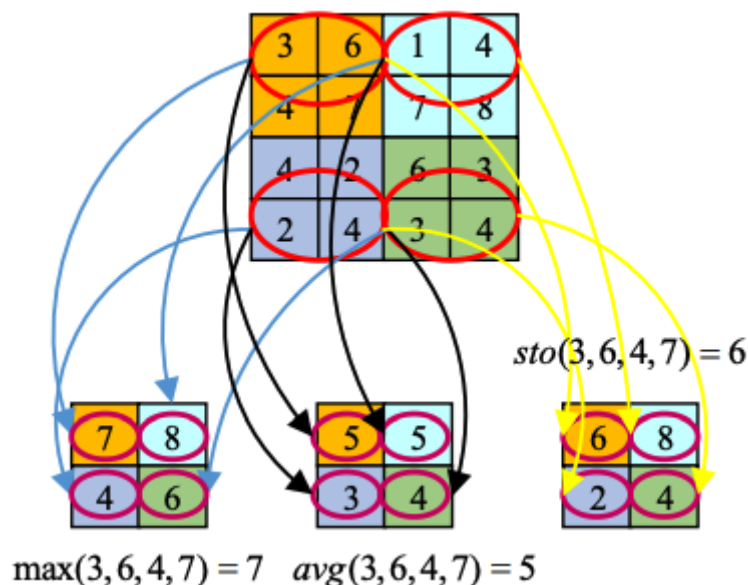
2.4.2.2 Združevalni sloj

V konvolucijske nevronske mreže se pogosto vstavljajo tudi združevalni sloji (ang. pooling layer), ker s svojo uporabnostjo v veliki meri zmanjšajo količino računskih operacij. Imenuje se lahko tudi proces podvzorčenja, ki zmanjša dimenzijo matrike značilk ter preprečuje preveliko prilaganje učnim podatkom (ang. overfitting).

Koncept je podoben konvoluciji, s pomikanjem po širini in višini vhodne matrike, vendar namesto transformacije kot v konvoluciji, pride do združenja po eni izmed izbranih metod. Najpogosteje se uporabljajo maksimalno podvzorčenje (ang. max pooling), povprečno podvzorčenje (ang. average pooling) in naključno podvzorčenje (ang. random pooling). Slika 13 prikazuje vse primere, kjer se iz lokalnih vzorcev vhodnega tenzorja pridobi izhodni tenzor v pomanjšani dimenziji.

S premikanjem okna poljubne velikosti, čez celotno sliko, se na vsaki točki izbere največjo vrednost, ostale vrednosti so zanemarljive. Običajno se uporabljajo okna velikosti 2×2 ali 4×4 . Korak premikanja je enak velikosti okna, da se vrednosti ne podvajajo.

Poznamo še globalno povprečno podvzorčenje, ki je tip ekstremnega podvzorčenja, kjer matriko atributov, oziroma tenzor, z dimenzijo širina $\times$ višina podvzorči v 1×1 . Pomeni, da vzame povprečje vseh elementov v matriki atributov, pri čemer se globina ohrani. Običajno se ta operacija izvede le pred polno povezovalnim slojem. Metoda ima dve prednosti, da zmanjša število učnih parametrov in omogoči konvolucijski nevronske mreži, da sprejme vhodne podatke sprejemljive velikosti (Lin et al., 2013).



Slika 13: Procesi podvzorčenja; max - maksimalno podvzorčenje, avg - povprečno podvzorčenje in sto - naključno podvzorčenje (Tian, 2020)

2.4.2.3 Polno povezani sloj

Zadnji, izhodni sloj iz konvolucije ali podvzorčenja je transformiran v števila ali vektorje in je povezan v polno povezani sloj tako, da je vsak vhodni podatek povezan z rezultatom z učenno utežjo. Na ta način je jasno, kateri piksli bodo dodeljeni posameznim razredom. Izhodni podatek je verjetnostna porazdelitev nad klasifikacijskimi razredi. Končni polno povezani sloj ima tipično takšno število različnih vrednosti, kot je definiranih razredov v klasifikaciji.

Po zadnjem polno povezanim sloju se običajno doda še aktivacijska funkcija, ki se razlikuje od prej naštetih. Glavni namen pri klasifikaciji je normalizacija vrednosti tako, da se porazdelijo med 0 in 1.

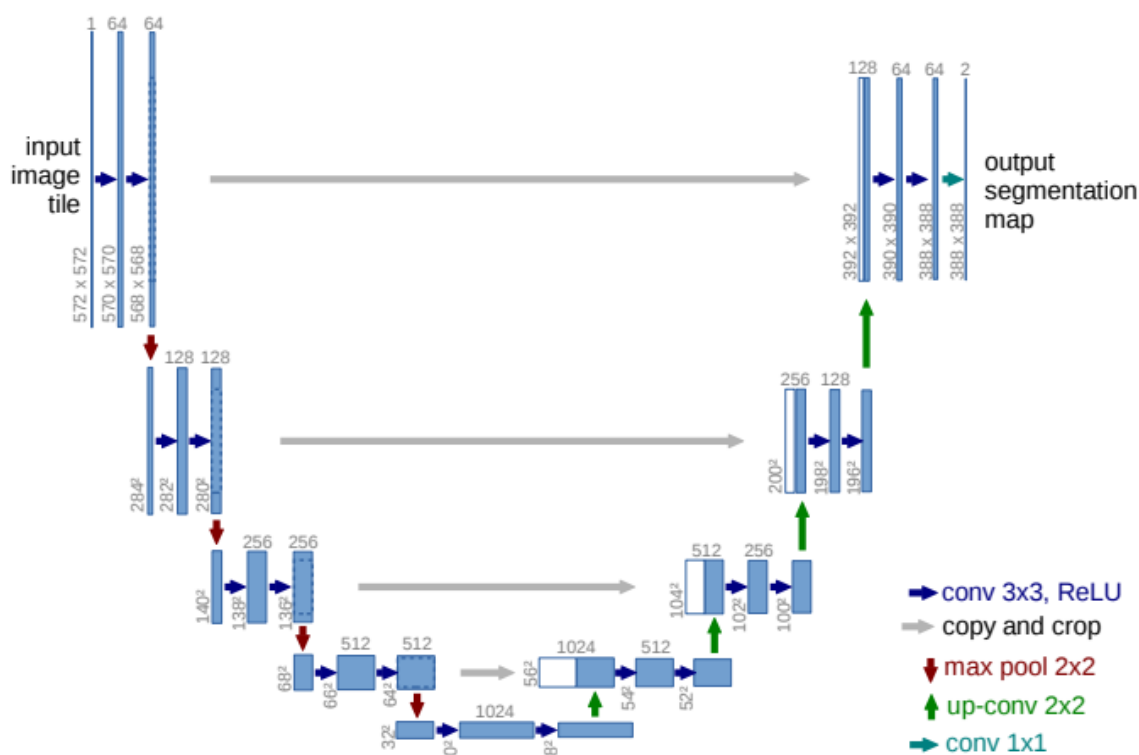
2.4.3 Primeri zgrajenih arhitekturnih modelov CNN

2.4.3.1 Osnovna arhitektura U-net

U-net nevronska mreža je bila prvotno namenjena za biomedicinske namene v semantični segmentaciji. Osnovni princip temelji na kodiranju vhodnih in dekodiranju procesiranih podatkov. Za razliko od klasifikacije, kjer rezultat predstavlja željen razred ali verjetnost razreda za izbrano območje, pri semantični segmentaciji dela na nivoju piksla. Lastnosti, ki so potrebne za vsak piksel, se model nauči v procesu kodiranja, ki jih nato preslika v nivo piksla, da dobimo končen rezultat (Ronneberger et al., 2015).

Kodiranje (ang. encoder) predstavlja levo stran grafa na sliki 14. Kot je razvidno, sestavlja ponavljajoče dvojne 3 x 3 konvolucije, katerim sledi ReLU in paketna normalizacija. Nato je uporabljen združevalni sloj maksimalnega podvzorčenja dimenzije 2 x 2. V vsakem bloku se prostorska dimenzija razpolovi in globinska dimenzija podvoji. Običajno je kodirnik sestavljen iz več nivojev. Kot primer kodirnikov lahko omenimo ResNet, MobileNetV2, EfficientNet in mnoge druge.

Dekodiranje (ang. decoder) je prikazano na desni polovici na grafu slike 14. Če se želi iz matrik značilk pridobiti željeno vrednost, torej en rezultat, je dekodiranje nujno potrebno. V vsakem koraku vzorčimo matriko značilk, kateri sledi 2 x 2 transponirana konvolucija, ki zmanjša globino oziroma število kanalov. Uporabi se tudi konvolucija 3 x 3, kateri sledi ReLU. Za zadnji sloj se uporabi polno povezani sloj, to je 1 x 1 konvolucija, ki pretvori podatke v željeno število kanalov oziroma razredov.



Slika 14: Arhitektura U-net-a. Modri štirikotniki predstavljajo matriko značilk z več kanali. Število kanalov je zapisano nad modrimi štirikotniki, medtem ko je dimenzija zapisana na spodnji levi strani le teh. Beli štirikotniki izražajo kopirane matrike značilk. (Ronneberger et al., 2015)

2.4.3.2 MobileNetV2

Opredelitev terminologije pred opredelitvijo arhitekture modela:

1. Globinsko ločljive konvolucije (ang. Depthwise Separable Convolutions, DSC):

Glavni namen je optimizirati arhitekturo modela tako, da se standardne konvolucijske sloje razdeli na dva ločena sloja. Prvi sloj se imenuje globinska konvolucija, ki opravi filtriranje podatkov glede na en filter po globini. Drugi sloj se imenuje točkovna konvolucija, ki ima velikost 1×1 in je odgovorna za izgradnjo novih lastnosti.

Pri standardni konvoluciji vhodni tenzor L_i , dimenzije $h_i \times w_i \times d_i$, ob filtru $K \in R^{k \times k \times d_i \times d_j}$, vrne izhodni tenzor L_j . Pri tem je količina računanja

$$h_i \times w_i \times d_i \times d_j \times k \times k \quad (8)$$

in nakazuje mero količine virov, ki jih nevronska mreža potrebuje za učenje. Mera količine virov je pomembna informacija, da vemo koliko časa in računalniške zmogljivosti se potrebuje pri učenju. Globinsko ločljive konvolucije, z zmanjšano količino računanja, delujejo skoraj tako dobro kot standardne:

$$h_i \times w_i \times d_i(k^2 + d_j), \quad (9)$$

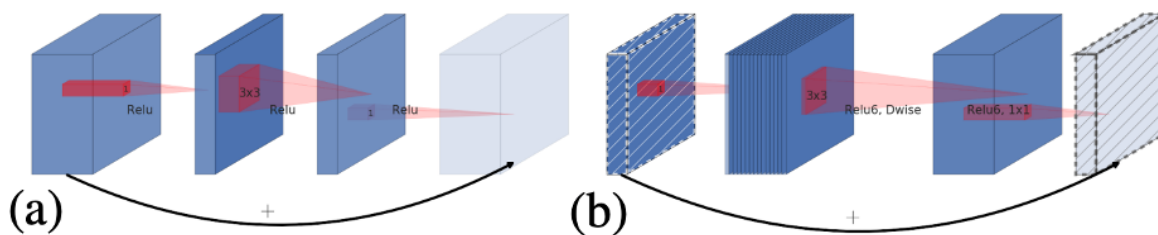
ki deluje kot seštevek obeh slojev. Z uporabo globinske ločljive konvolucije je količina računanja, oziroma število potrebnih operacij, kar 8 do 9 krat manjša (Sandler et al., 2018).

2. Linearni zastoji (ang. linear bottlenecks)

Izpostavimo dve lastnosti, in sicer neničeln rezultat sloja v formatu ReLU(Bx). Ujemajoči del vhodnega prostora (x) omeji na linearno transformacijo (Bx). Globoka nevronska mreža je omejena na linearno klasifikacijo v neničelnem delu izhodne domene. Druga lastnost pravi, da kadar uporabimo ReLU, lahko odpove kanal, oziroma se lahko poruši sloj, kar vodi do izgube podatkov v kanalu. Ko je prisotno veliko število kanalov, se temu izognemo, saj se podatki shranijo v nižjo-dimenzijski podprostor in se ob uporabi ReLU podatki ne izgubijo (Sandler et al., 2018).

3. Obrnjeni ostanki (ang. inverted residuals)

Zastojni bloki (ang. bottleneck blocks) so podobni preostalim blokom, kjer vsak blok vsebuje vhodni podatek. Vhodnemu podatku sledijo zastojni bloki in nato razširitev do velikosti vhodnih podatkov (svetleje obarvan del na sliki 15). Dodane so bližnjice, med različnimi zastojnimi bloki, da izboljšajo sposobnost gradientnega širjenja med različnimi sloji. Obrnjeni način je bolj procesno učinkovit in v nekaterih nalogah deluje nekoliko bolje. Slika 15 prikazuje razlike med običajnimi bloki (a) in obrnjenimi bloki (b) (Sandler et al., 2018).



Slika 15: a) blok ostankov, b) blok obrnjenih ostankov (Sandler et al., 2018)

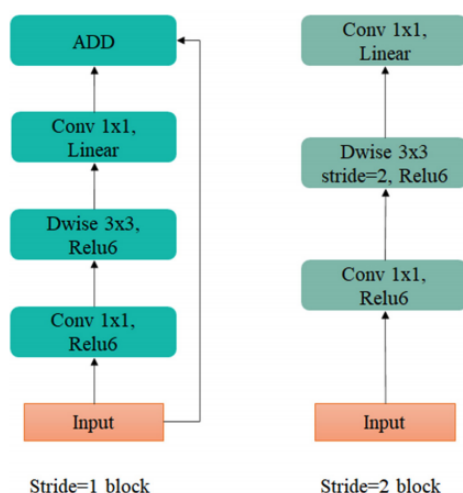
Količino računanja in število parametrov zastoje konvolucije se izračuna z enačbo:

$$h \times w \times d_i \times t(d_i + k^2 + d_j), \quad (10)$$

kjer $h \times w$ predstavlja velikost bloka, d_i število vhodnih kanalov, t faktor razširjanja, d_j število izhodnih kanalov in k velikost filtra. Primerjava z enačbo (9), prikaže, da je v tem primeru dodatni element, ki je potreben zaradi dodatne 1×1 konvolucije.

Sandler et al. (Sandler et al., 2018) razlaga, da vsak blok v modelu MobileNetV2 vsebuje tako 1×1 razširitveni sloj, kot tudi globinski in točkovni sloj. Točkovni konvolucijski sloj se imenuje tudi projekcijski sloj, ki preslika podatke z velikim številom kanalov v tenzor, z veliko manjšim številom kanalov. Zastojni blok ostankov vsebuje izhodne podatke vsakega zastojnega bloka. Pri tem se izpostavi, da razširitveni konvolucijski sloj poveča število kanalov glede na razširitveni faktor, ki je definiran pred globinsko konvolucijo. Vsak sloj ima paketno normalizacijo (v eksperimentalnem primeru uporabim individualno normalizacijo) in ReLU6, kot aktivacijsko funkcijo (Sandler et al., 2018).

Model vsebuje skupno 17 zastojnih blokov ostankov, ki pa se lahko razlikujejo glede na cilj naloge. Sledi še standardna 1×1 konvolucija, globalno povprečno podvzorčenje in končni klasifikacijski sloj. Dodan je tudi shematični prikaz (Slika 16) (Michele et al., 2019).



Slika 16: Konvolucijski bloki v MobileNetV2. Leva del prikazuje zaporedje konvolucijskih slojev s korakom premikanja filtra 1 piksel. Desna del pa navaja zaporedje virov, s korakom premikanja filtra za 2 piksla. Dwise – globinsko ločljivi konvolucijski sloj, Conv – konvolucijski sloj. Vhod (ang. input) in izhod se na koncu združita (ADD), za kar je potreben konvolucijski sloj. (Dong et al., 2020)

2.4.3.3 EfficientNet

EfficientNet je novejši model, ki ga je razvil Google (Tan & Le, 2019) za izboljšanje CNN modelov. Temelji na prilagajanju merila. Uporablja preprost, zelo zmogljiv sestavljeni koeficient (ang. compound coefficient). Deluje na principu enakomernega spreminjanja vseh dimenzij, torej širine, globine in ločljivosti, s fiksnim nizom koeficientov prilagajanja, različno od tradicionalnih metod, ki posamezno merijo dimenzije omrežij. Spreminjanje merila posameznih dimenzij izboljša zmogljivost modela, uravnoteži vse dimenzije, s koeficientom prilagajanja pa izboljša zmogljivost celotnega delovanja (slika 17). Osnovno omrežje zelo vpliva na učinkovitost spreminjanja merila. To se uredi z uporabo orodja AutoML, ki optimizira in precizira zmogljivost (FLOPS⁴ – Floating point operations per second), saj ustvari novo osnovno omrežje za iskanje arhitekture modela.

Avtomatsko strojno učenje (ang. Automated Machine Learning – AutoML) pomeni avtomatizacija grajenja strukture nevronske mreže. Preko inteligentnega arhitekturnega manipuliranja AutoML naredi globoko učenje bolj dostopno za uporabnike in pospeši razvoj globokega učenja.

Podobno kot pri modelu MobileNetV2 tudi EfficientNet uporablja obrnjeno zastojno konvolucijo (ang. mobile inverted bottleneck convolution - MBConv) kot glavni gradnik, oziroma blok. Z razliko od

⁴ FLOPS (ang. floating point operations per second) nam pove število opravljenih matematičnih operacij s plavajočo vejico na sekundo.

večine modelov, ki uporabljajo ReLU aktivacijsko funkcijo, EfficientNet uporablja novejšo aktivacijsko funkcijo Swish, ki je boljše od ReLU, glede na raziskave (Ramachandran et al., 2018) in (Ramachandran et al., 2017).

Tan in Le v (Tan & Le, 2019) izpostavita dvojno opažanje:

1. S spreminjanjem merila katerekoli dimenzije mreže, torej širine, globine ali ločljivosti, se izboljša natančnost modela, pri zmanjšanju napredka za večje modele.
2. V koliko se želi izboljšati natančnost in zmogljivost modela, je ključnega pomena, da se uravnotežijo vse dimenzije mreže, kar pomeni najdeno najboljšo kombinacijo med globino, širino in ločljivostjo.

Glede na opazovanja sta predlagala novo kombinirano spreminjanje velikosti dimenzij, ki uporablja sestavljeni koeficient Φ , da uravnoteženo prilagodi velikosti vseh dimenzij po principu:

$$\text{globina: } d = \alpha^\Phi,$$

$$\text{širina: } w = \beta^\Phi \text{ in}$$

$$\text{ločljivost: } r = \gamma^\Phi,$$

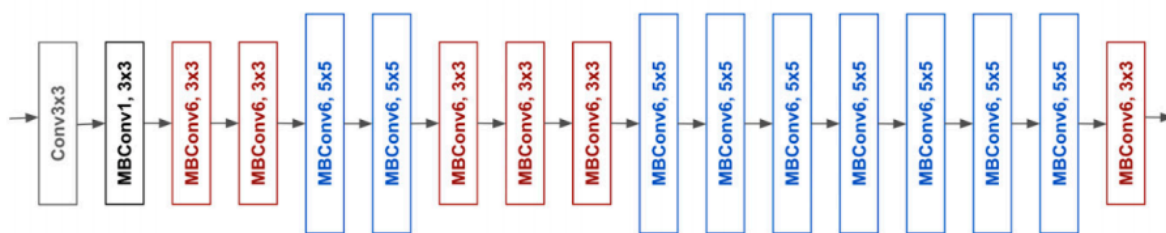
kjer so α, β, γ konstante in Φ koeficient, ki ga določi uporabnik ob predpostavki:

$$\alpha \times \beta^2 \times \gamma^2 \approx 2 \text{ in} \quad (11)$$

$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1.$$

Glede na ta koeficient (Φ) ločujemo različne stopnje EfficientNet-a.

Dodan je shematični prikaz arhitekture modela EfficientNet (slika 17) (Tan & Le, 2019)



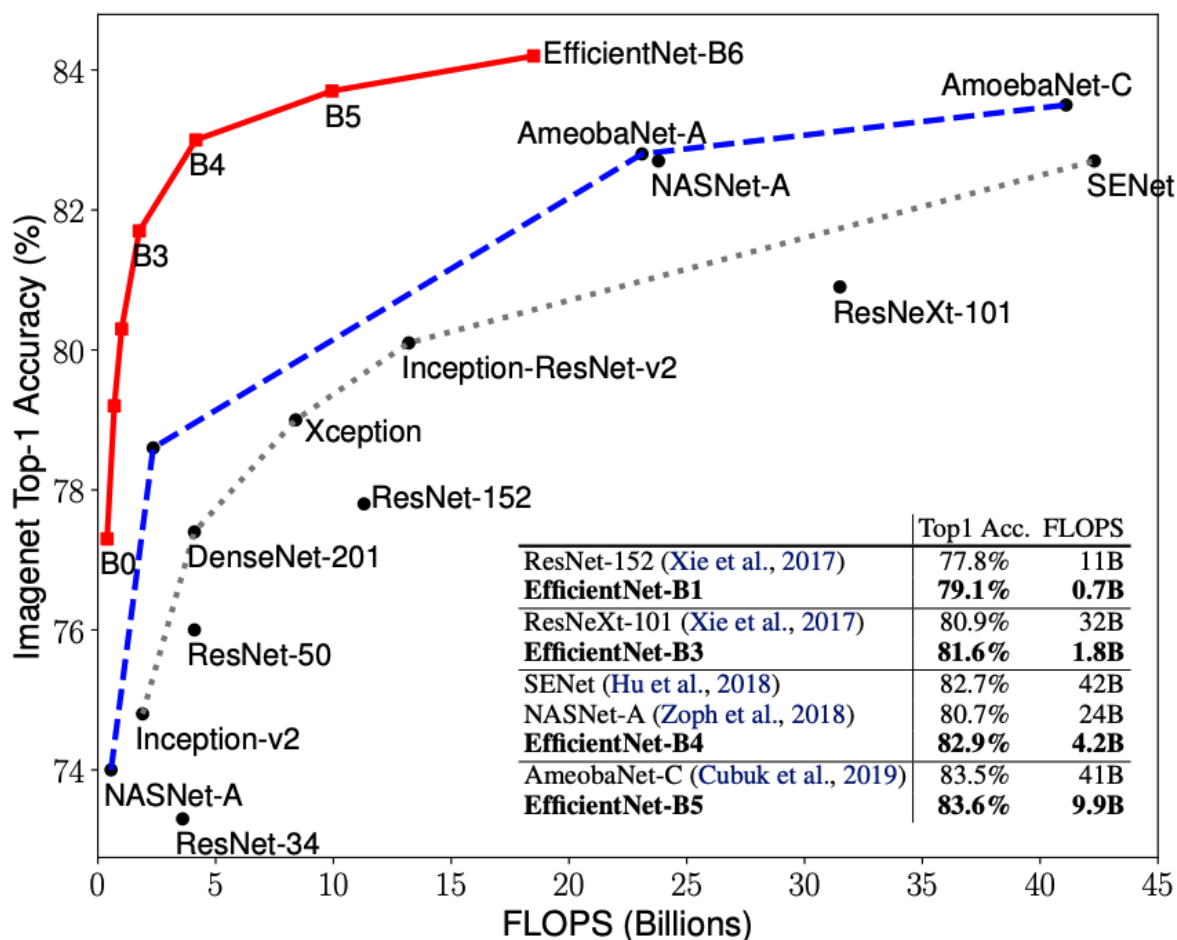
Slika 17: Shematični prikaz osnove arhitekture EfficientNet-B0. MBConv – obrnjena zastoja konvolucija in Conv – konvolucija. (Bazi et al., 2019)

Ko je pripravljeno izhodišče EfficientNet-B0, se uporabi metoda kombiniranega spreminjanja merila dimenzij, da se pridobi EfficientNet-B1 do B7. Postopek se opravi v dveh korakih:

1. Najprej z nastavitvijo koeficient $\Phi = 1$ ob pogoju, da je dovolj podatkov, prilagodimo konstante, s previdnostjo na pravilo $\alpha * \beta^2 * \gamma^2 \approx 2$. Izračun se naredi na izhodiščnem EfficientNet-B0.

2. Ko imamo pripravljene konstante, jih uporabimo v celotnem nadaljevanju, pri čemer spreminjamo le koeficient Φ , da pridobimo naslednje stopnje B1 do B7.

V raziskavi (Tan & Le, 2019) se je EfficientNet izkazal za izredno zmogljivega in natančnega v primerjavi z obstoječimi modeli. Tudi model Gpipe, ki je nedavno veljal za najboljšega in procesno najbolj potratnega, je bil EfficientNet-B7 dokazan s primerljivo natančnostjo in veliko boljšo zmogljivostjo. V testnem primeru gre za kar 8,4-krat manjšo porabo parametrov. Slika 18 prikazuje, kako so modeli razporejeni po natančnosti in zmogljivosti (FLOPS).



Slika 18: Grafični prikaz razporeditve modelov glede na natančnost in zmogljivost (FLOPS) (Tan & Le, 2019)

2.4.4 Hiperparametri in fine nastavitve

2.4.4.1 Optimizator

Poznamo veliko vrst optimizatorjev, omenili bomo dva, ki se uporabljata v eksperimentalnem delu. Na splošno se ti tehniki optimizacije, ki se najpogosteje uporabljata in sta najbolj prilagodljivi za širok spekter nalog.

- RMSprop – Root Mean Square Propagation

RMSprop je na področju algoritmov neobjavljen algoritem optimizacije, vendar med poznavalci strojnega učenja zelo znan in pogosto uporabljen. Predstavil ga je Geoffrey Hinton (Tieleman & Hinton, 2012). Gradienti zelo zapletenih nevronske mrež se nagibajo k izginotju, ko se podatki širijo po funkciji (problem izginjajočih gradientov). RMSprop je bil razvit kot stohastična tehnika za mini-paketno učenje (ang. mini-batch learning).

Zgoraj omenjen problem, RMSprop rešuje z uporabo premikajočih povprečij kvadratnega gradienta, da normalizira gradient. Normalizacija uravnovesi velikost koraka (momentum), kar pomeni, da zmanjša korak za velike gradiente in poveča korak za manjše gradiente. Izračuna se po enačbah:

$$v_t = \rho v_{t-1} + (1 - \rho) \times g_t^2, \quad (12)$$

$$\Delta \omega_t = -\frac{\eta}{\sqrt{v_t + \epsilon}} \times g_t \text{ in} \quad (13)$$

$$\omega_{t+1} = \omega_t + \Delta \omega_t, \quad (14)$$

kjer je η začetna stopnja učenja, v_t eksponentno povprečje kvadratnega gradienta in g_t gradient v času t z ω . Hiperparameter ρ je običajno velikosti 0,9 in ϵ vrednost 1e-10, da se izognemo deljenju z 0.

- Adam

Adam je metoda za učinkovito stohastično optimizacijo, ki zahteva le gradient prvega reda z nizko količino zahtevanega računalniškega spomina. Metoda izračunava posamezne stopnje učenja za različne parametre iz ocenjene vrednosti gradienta. Nastal je z združenjem dveh pristopov stohastičnega gradientnega spusta, prilagodljivega gradienta (ang. adaptive gradient) in RMSprop. Za izračun, namesto dejanskega spusta za celoten niz podatkov, optimizacijski algoritem, uporabi naključno izbrano podatkovno množico za ustvarjanje stohastičnega približka (Kingma & Ba, 2014).

Nekatere prednosti algoritma Adam so, da je velikost posodobitev parametrov nespremenjena s spreminjanjem velikosti gradienta. Velikost koraka je približno omejena z velikostjo hiperparametra.

Izračuna se ga v enačbah:

$$v_t = \beta_1 v_{t-1} + (1 - \beta_1) \times g_t, \quad (15)$$

$$s_t = \beta_2 s_{t-1} + (1 - \beta_2) \times g_t^2, \quad (16)$$

$$\Delta\omega_t = -\eta \frac{v_t}{\sqrt{s_t + \epsilon}} \times g_t \text{ in} \quad (17)$$

$$\omega_{t+1} = \omega_t + \Delta\omega_t, \quad (18)$$

kjer je η začetna stopnja učenja, v_t eksponentno povprečje gradienta po ω , s_t eksponentno povprečje kvadratnega gradienta po ω in g_t gradient v času t z ω . Hiperparameter β_1 je običajno velikosti 0,9 in β_2 velikosti 0,99. Vrednost ϵ je običajno $1e-10$, da se izognemo deljenju z 0.

2.4.4.2 Stopnja učenja

Globoke nevronske mreže se uči z uporabo stohastičnega gradientnega spusta. To je iterativna metoda za optimiziranje funkcije z dodano lastnostjo glajenja. Stopnja učenja je optimizacijski algoritem za oceno napake gradienta na učni množici, ki popravi uteži z algoritmom vzratnega razširjanja napake.

Velikost spremembe uteži med učenjem modela se imenuje velikost koraka ali stopnja učenja (ang. Learning Rate). Je prilagodljiv hiperparameter, ki se uporablja za učenje globokih nevronske mrež in predstavlja pozitivno vrednosti med 0 in 1 (Goodfellow et al., 2016) (str. 290–293).

Stopnja učenja določa, v kolikšnem času se bo model prilagodil določenim problemom. Kadar je stopnja učenja velika, pride do hitrih sprememb na utežeh z manjšim številom epoh. Obratno velja za manjšo vrednost stopnje učenja. Problem nastane, če je vrednost prevelika, kar lahko prikaže navidezno optimalno rešitev, medtem ko se pri premajhni vrednosti model zatakne. Iz tega razloga v (Goodfellow et al., 2016) (str. 429–430) afirmirajo, da je optimizacija stopnje učenja najpomembnejša med hiperparametri.

2.4.4.3 Kriterijska funkcija

V procesu optimizacije je funkcija, ki opredeli kandidata (rezultat), objektivna funkcija. Z maksimizacijo ali minimizacijo funkcije pomeni, da ima izbrani kandidat najvišjo ali najnižjo verjetnost. Pri nevronske mrežah se maksimizira. Dobila je ime kriterijska funkcija (ang. loss function). Vrednost, ki je rezultat te funkcije, je izguba (ang. loss).

Opravlja pomembno vlogo, ker združuje vse vidike modelov eno skalarno vrednost. Ob znižanju vrednosti skalarja se da razbrati, da je izboljššan tudi model.

Razvitih je bilo že veliko različic funkcij izgube, omenil bom dve, ki sta bili uporabljeni v eksperimentalnem delu:

- Binarna prečna entropija (ang. Binary Cross Entropy, BCE)

Informacije količinsko opredeljujejo število bitov, potrebnih za kodiranje in prenos dogodka. Dogodki z manjšo verjetnostjo vsebujejo več informacij in dogodki z večjo verjetnostjo vsebujejo manj informacij.

Entropija je število bitov, ki so potrebni za prenos naključno izbranega dogodka iz verjetnostne porazdelitve. Porazdelitev, kjer imajo dogodki enako verjetnost ima večjo entropijo in obratno.

Prečna entropija temelji na ideji entropije iz teorije informacij in izračuna število bitov, ki so potrebni za predstavitev ali prenos povprečnega dogodka iz ene porazdelitve v primerjavi z drugo porazdelitvijo.

Vsak primer ima znano oznako razreda z verjetnostjo 1,0 in verjetnostjo 0,0 za vse druge oznake. Model lahko oceni verjetnost primera, ki pripada vsaki oznaki razreda. Razvrstitev enega primera se preslika na naključno spremenljivko z verjetnostno porazdelitvijo kot:

- naključno spremenljivko – kjer se potrebuje napovedano oznako razreda, ali kot
- dogodek – vsako oznako razreda je mogoče predvideti.

Iz tega se lahko izlušči definicija prečne entropije – je mera razlike med dvema verjetnostnima porazdelitvama za naključno spremenljivko ali set dogodkov. Izračuna se kot (Jadon, 2020):

$$BCE = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})), \quad (19)$$

kjer je $\hat{y}$ napovedana verjetnost in y pričakovana verjetnost.

- izguba Dice (ang. Dice loss – DL)

Izguba Dice temelji na koeficientu Dice, ki meri prekrivanje med dvema vzorcema. Mera je velikosti od 0 do 1, kjer 1 pomeni popolno in celotno prekrivanje. Koeficient Dice je bil prvotno razvit za binarne podatke in ga lahko izračunamo kot (X. Li et al., 2019):

$$Dice = \frac{2|A \cap B|}{|A| + |B|} = \frac{2 \sum_{piksl} y_{pravi} y_{napovedani}}{\sum_{piksl} y_{pravi}^2 + \sum_{piksl} y_{napovedani}^2}, \quad (20)$$

kjer je $|A \cap B|$ presek med A in B, medtem ko $|A|$ definira vsoto elementov v množici A (enako velja za B).

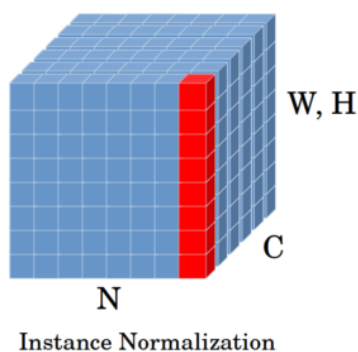
2.4.4.4 Normalizacija

Med učenjem nevronske mreže se velikokrat uporabljajo podatki, ki se med seboj razlikujejo po intervalu vrednosti in ti podatki, bodo vplivali na kakovost in učinkovitost učenja. Brez normalizacije nekatere robne lastnosti v večjih intervalih ne bi bile upoštevane. Normalizacija se običajno uporablja na začetku, ker se operira še s surovimi podatki. Pri globokih nevronske mrežah, lahko v skritih slojih, pride do velikih razhajanj v vrednostih, za kar so potrebne tudi vmesne normalizacije.

Najpogosteje uporabljena normalizacija je paketna normalizacija (ang. batch normalization), ki hkrati upošteva celotno matriko atributov enega kanala. Ulyanov et al. v raziskavi (Ulyanov et al., 2016) opredelijo primerjavo paketne normalizacije in normalizacije primerka, z ugotovitvijo, da z normalizacijo primerka izboljšamo učinkovitost učenja globoke nevronske mreže.

Normalizacija primerka (ang. instance normalization) se imenuje tudi kontrastna normalizacija, ki deluje na enem vzorcu in normalizira kontrast med prostorskimi elementi vzorca. CNN opravi tudi normalizacijo intenzitete po višini in širini na eni matriki atributov enega primera.

Z definicijo, kako deluje normalizacija primerka, se opredeli x kot vhodni tenzor, ki ga sestavlja serija slik N . Vsaka slika ima C kanale z višino H in širino W , zato je $x \in \mathbb{R}^{N \times C \times H \times W}$ štiri-dimenzionalen tenzor. Učni primer – tenzor (rdeče označen na sliki 19). Izračuna se srednja vrednost (21) in varianco (22) preko prostorske lokacije (W in H) (Ulyanov et al., 2017).



Slika 19: Prikaz vhodnih podatkov v normalizacijo primerka po dimenzijah (Karimi, 2021)

Izračun srednje vrednosti in variance za primer x_{ni} pri normalizaciji primerka:

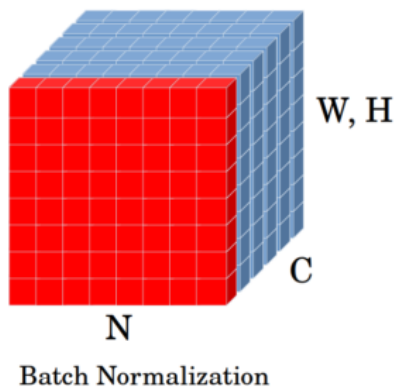
$$\mu_{ni} = \frac{1}{HW} \sum_{l=1}^W \sum_{m=1}^H x_{nilm} \quad (21)$$

$$\sigma_{ni}^2 = \frac{1}{HW} \sum_{l=1}^W \sum_{m=1}^H (x_{nilm} - \mu_{ni})^2. \quad (22)$$

Rezultat normalizacije je:

$$y_{nijk} = \frac{x_{nijk} - \mu_{ni}}{\sqrt{\sigma_{ni}^2 + \epsilon}}, \quad (23)$$

kjer je ϵ majhna vrednost za boljšo stabilizacijo modela. Mreža se pri tem nauči dva parametra (γ in β). Paketno normalizacijo se kvalificira na podoben način kot normalizacijo primerka, z opredelitvijo enega kanala, oziroma matriko atributov čez vse učne vzorce v mini paketu (rdečo na sliki 20). $B = \{x_1, x_2, \dots, x_N\}$ se definira kot mini paket. Z uporabo srednje vrednosti (μ_{ni}) in variance (σ_{ni}^2) mini paketa se normalizira vrednosti, pri čemer se mreža z gradientnim spustom nauči dva parametra (γ in β).



Slika 20: Prikaz vhodnih podatkov v paketni normalizaciji po dimenzijah (Karimi, 2021)

Izračun srednje vrednosti in variance za primer x_i pri paketni normalizaciji:

$$\mu_i = \frac{1}{NHW} \sum_{n=1}^N \sum_{l=1}^W \sum_{m=1}^H x_{nilm} \quad (24)$$

$$\sigma_i^2 = \frac{1}{NHW} \sum_{n=1}^N \sum_{l=1}^W \sum_{m=1}^H (x_{nilm} - \mu_i)^2 \quad (25)$$

$$\hat{x}_{nijk} = \frac{x_{nijk} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}} \quad (26)$$

Rezultat normalizacije:

$$y_{nijk} = \gamma \hat{x}_{nijk} + \beta \quad (27)$$

Obe normalizaciji pospešita učenje nevronske mreže. V uporabljenih modelih je vgrajena paketna normalizacija. Za postopek nadzorčenja sem uporabil normalizacijo primerka.

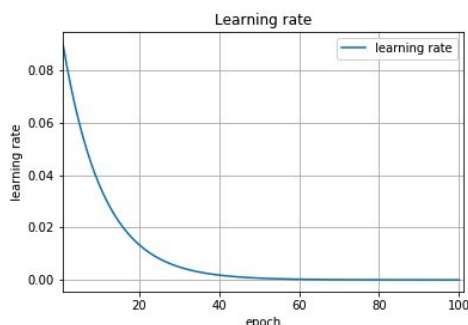
2.4.4.5 Zmanjšanje stopnje učenja

Tehnika zmanjševanja stopnje učenja (ang. Learning Rate Decay, LRD) je značilna za uporabo sodobnih nevronske mreže. Učenje se začne pri veliki stopnji, s postopnim zmanjševanjem, dokler ne doseže lokalnega minimuma. Pogosto se uporabljajo časovno osredotočeno zmanjševanje (ang. time-based decay), koračno zmanjševanje (ang. step decay) in eksponentno zmanjševanje (ang. exponential decay). Uporabil sem le eksponentno, zato bom opredelil le tega.

Eksponentno zmanjševanje stopnje učenja ima matematično enačbo:

$$lr = lr_0 \times e^{-kt}, \quad (28)$$

kjer je k hiperparameter in t številka iteracije. Vrednost je vedno manjša od 1. Spodnja slika poda grafični prikaz te funkcije.



Slika 21: Prikaz zmanjševanja stopnje učenja po epohah

2.4.5 Izračun števila parametrov

Pri konvolucijskih nevronske mrežah se pričakuje, da se model nauči vrednosti filtrov z uporabo vzratnega razširjanja napake. Število parametrov, ki ga izračunamo za vsak sloj in predstavljajo število učljivih elementov za filter. Splošno, gre za uteži, ki se jih model nauči in ki povedo kako dobra je zmogljivost napovedovanja modela. Spreminjajo se glede na vhodne podatke, optimizacije, ipd. Za vsak sloj, na različen način preračunavamo število parametrov:

1. vhodni sloj: namenjen je le kot vhod za slikovno dimenzijo,
2. konvolucijski sloj: izračuna se glede na enačbo

$$(m \times n \times d + 1) \times k, \quad (29)$$

kjer je $m \times n$ dimenzija filtra, d globina vhodnega podatka, k globina izhodnega podatka in 1 zaradi pristranskosti vsakega filtra,

3. združevalni sloj – namenjen je le za zmanjševanje dimenzije slike, zato nima parametrov in
4. polno povezovalni sloj:

$$(n + 1) \times m \quad (30)$$

Pogostokrat se pojavijo tudi neučni parametri, ki so pojav sloja paketne normalizacije, saj so posodobljeni s srednjo vrednostjo in varianco, vendar niso naučeni z vzratnim razširjanjem napake.

2.4.6 Prekomerno prileganje in tehnike bogatenja količine podatkov

Namen naučenega modela je optimalno delovanje na učni in neučeni množici podatkov, na katerih se izvajajo napovedi. Od modela se ne potrebuje samo učenja iz učnih primerov, ampak tudi generalizacijo za nove podatke. Najti pravo mero med naučenostjo in generalizacijo je zahtevna naloga.

Prihaja lahko do dveh skrajnosti:

1. Prava skrajnost je premalo naučen model, ki ne bo ustrezno prepoznal pravih podatkov na učni množici in bo pomanjkljivo napovedoval vrednosti na novih podatkih.
2. Druga skrajnost je preveč naučen model, ki bo optimalno prepoznal pravilne podatke na učni množici in pomanjkljivo napovedal vrednosti na novih podatkih.

Problem premalo naučenega modela se rešuje s spreminjanjem arhitekture modela, z dodajanjem kompleksnejše funkcije, slojev in povečanjem kapacitete, oziroma kompleksnosti modela. Ker je rešitev dokaj enostavna, se tovrstna težava redko pojavi, nasprotno pa je problem preveč naučenega modela zelo pogost.

Preveč naučenega modela ali prekomernega prileganja učni množici se preprosto spremlja med učenjem, z izrisovanjem linijskih grafov učne in validacijske množice (običajno vrednost izgube). V primeru tega pojava se model rešuje na dva načina:

1. Prvi je redukcija prekomernega prileganja, z zmanjšanjem kompleksnosti nevronske mreže. Rešitev leži v spremembi mrežne strukture (števila uteži) ali parametrov mreže (vrednosti uteži).
2. Drug je redukcija prekomernega prileganja, s povečanjem učne množice. Dodatno se lahko zajame podatke, v kolikor niso dostopni se uporabijo tehnike bogatenja količine podatkov.

Za bogatenje količine podatkov se uporabljajo tehnike za ustvarjanje umetnih posnetkov s spreminjanjem originalnih posnetkov s transformacijami, rotacijami, spreminjanjem svetlosti, kontrasta, dodajanjem šuma ali zamegljenosti ipd. Za vsako tehniko se lahko izbere naključno ali načrtovano spreminjanje originalnih posnetkov. Način bogatenja je odvisen od specifične domene (npr. v kolikor so prisotni prometni znaki, se slike ne sme poljubno rotirati). Bogatenje količine podatkov se običajno dogaja naključno v vsaki epohi.

2.5 Matrike kakovosti naučenih modelov

Ocenjevanje modelov stojnega učenja ima ključno vlogo pri vsakem projektu. Model je mogoče ocenjevati na več načinov, npr. preko natančnosti. Vendar, tudi ob dobri oceni natančnosti ni zagotovljeno, da je model dobro zgrajen, zato se uporabijo še druge metrike. V nadaljevanju izpostavljam najpogostejše uporabljene.

2.5.1 Matrika zamenjav za binarno klasifikacijo

V binarni klasifikaciji so vhodni podatki razdeljeni v dva razreda. Običajno so označeni kot 1 in 0, ali kot pozitivne in negativne vrednosti. Na primeru prepoznavanja poplav iz satelitskih posnetkov teh vrednosti se ovrednoti ali je območje poplavljen ali ni.

Pomembno je, da imamo tudi numerično oceno oziroma verjetnost vsakega dela (npr. piksla), ki ga klasificiramo, da pripada določenemu razredu.

Za pridobitev informacije o uspešnosti modela se uporablja matrika zamenjav. Primer matrike zamenjav, za binarni problem, je viden na preglednici 3, kjer stolpci predstavljajo napovedane vrednosti in vrstice dejanske vrednosti. V obeh so prisotne tako pozitivne, kot negativne vrednosti. Tako se sestavi matrika, ki vsebuje štiri elemente (rdečo in zeleno) za predstavitev pravih in napačnih napovedi. Vrednosti, označene zeleno, predstavljajo pravilne napovedi, medtem ko rdeče označujejo napačne napovedi. Znale so štiri metrike v matriki zamenjav, in sicer:

- pravilne pozitivne (ang. True Positive – TP) – kolikokrat je model pravilno klasificiral pozitivne vrednosti,
- napačne negativne (ang. False Negative – FN) – kolikokrat je model napačno klasificiral negativne vrednosti,
- napačne pozitivne (ang. False Positive – FP) – kolikokrat je model napačno klasificiral pozitivne vrednosti in
- pravilne negativne (ang. True Negative – TN) – kolikokrat je model pravilno klasificiral negativne vrednosti.

Preglednica 3: Matrika zamenjav za binarno klasifikacijo

		NAPOVED	
		Pozitivno	Negativno
RESNICA	Pozitivno	TP	FN
	Negativno	FP	TN

Iz matrike zamenjav je mogoče izračunati štiri vrednosti za ocenitev modela:

- Prva je natančnost (ang. accuracy, ACC) – vse pravilno klasificirane vrednosti se delijo z vsoto vseh vrednosti.

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (31)$$

- Druga je točnost (ang. precision, PRE) – pravilno klasificirane pozitivne vrednosti se delijo z vsoto pravih vrednosti.

$$PRE = \frac{TP}{TP+FP} \quad (32)$$

- Tretja je občutljivost (ang. sensitivity ali recall, SEN) – meri sposobnost modela, da prepozna pozitivne vrednosti (npr. je poplava).

$$SEN = \frac{TP}{TP+FN} \quad (33)$$

- Četrta je specifičnost (ang. specificity, SPE) – meri sposobnost modela, da prepozna negativne vrednosti (npr. ni poplava).

$$SPE = \frac{TN}{TN+FP} \quad (34)$$

2.5.2 Območje pod krivuljo

Območje pod krivuljo AUC (ang. Area Under Curve) je način, s katerim se ugotovi povprečna natančnost modela. Vrednost variira od 0 do 1, kjer 1 predstavlja popolno natančen model. Izračuna se kot območje pod krivuljo ROC (ang. Receiver Operating Characteristic), ki predstavlja zvezo med občutljivostjo (SEN) in specifičnostjo (SPE).

Namesto preobremenitve z matriko zamenjav in zaradi lažje preglednosti se uporablja krivulja ROC. Ta izvira iz teorije zaznavanja signala. Zvezo med SEN in SPE za določen klasifikator je potrebno izrisati podobno kot na sliki 22. Na vertikalni osi so prave pozitivne vrednosti ali SEN, in na horizontalni osi so napačne pozitivne vrednosti ali 1 – SPE. Poznamo enotni klasifikator, ki napove samo en razred, kjer imamo le eno točko na ROC krivulji. V drugem načinu klasifikator vrne verjetnost, to je verjetnostni klasifikator. Za vsak x verjetnostni klasifikator vrne verjetnost $f(x, +)$ za pozitivne razrede in verjetnost $f(x, -)$ za negativne razrede. Znan je tudi še razvrstitveni klasifikator, ki ga definira izračun najverjetnejše pozitivne vrednosti x :

$$f(x) = \frac{f(x,+)}{f(x,-)} \quad (35)$$

Vrednost $f(x)$ se uporablja za razvrstitev primerov od najmanj do najbolj verjetne pozitivne vrednosti. Določiti se mora tudi prag, na katerem bo izračunana ROC vrednost. Najpogosteje se uporabi vrednost

praga 0,5 (Kononenko & Kuhar, 2007). V eksperimentalnem delu sem uporabil 200 pragov, ki so porazdeljeni od 0 do 1.

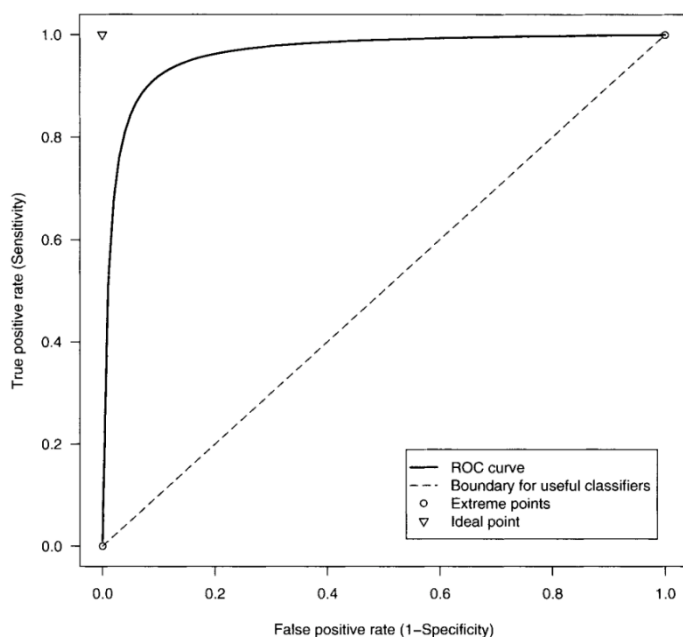
V kolikor je prag 0,0, to pomeni, da so vsi piksli pravilno klasificirani kot negativni in vsi pozitivni kot napačni. V tem primeru je SEN 0,0 in SPE 1,0 prikazan je spodnjem levem vogalu slike 22. Nasprotno velja za prag 1,0, kjer so vsi piksli pravilno klasificirani kot pozitivni in vsi negativni kot napačni. V drugem primeru sta SEN 1,0 in SPE 0,0, prikazana v zgornjem desnem vogalu na sliki 22. Linija, ki povezuje ti dve točki, predstavlja naključni klasifikator z uporabo različnih pragov. Idealno situacijo predstavlja točka v zgornjem levem vogalu na Slika 22, ki zavzame vrednosti SEN in SPE 1,0. Bolj se ROC krivulja približa idealni točki, boljši je model.

Krivuljo ROC se lahko uporabi tudi za analizo vrednosti napačno klasificiranih razredov.

Za analizo se želi dobiti numerične vrednosti, ki jih pridobimo z izračunom površine pod krivuljo ROC. Območje pod krivuljo AUC izračunamo z enačbo:

$$AUC = \int_0^1 SEN_T(1 - SPE)_T dT. \quad (36)$$

Večja kot je vrednost AUC, boljši bo model. Največja vrednost je 1, kar pomeni, da model popolnoma razlikuje med negativnimi in pozitivnimi vrednostmi.



Slika 22: Primer izrisa ROC krivulje in prikaz AUC (Kononenko & Kuhar, 2007)

AUC je v dobrem smislu zaželen iz dveh razlogov:

1. Namesto absolutnih vrednosti meri, kako dobra je napoved.
2. Meri kakovost napovedi modela ne glede na to, kateri prag klasifikacije je izbran.

2.5.3 Povprečno razmerje presek:unija - IoU

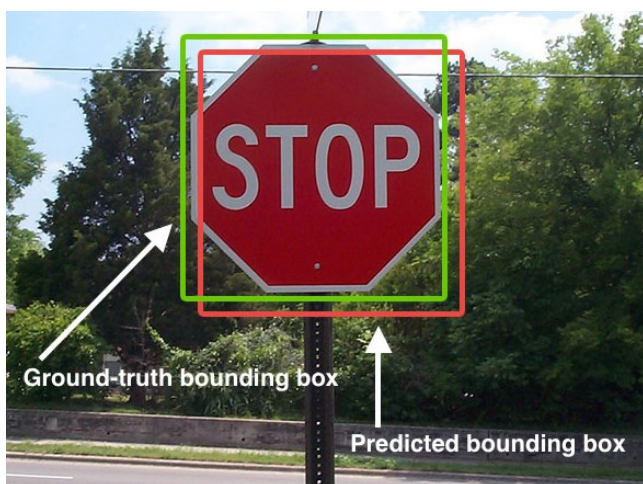
Povprečno razmerje presek:unija (ang. Intersection over Union - IoU) je metrika za ocenjevanje količine pravilno napovedanih pikslov, glede na pravilno označene vrednosti. Temelji na indeksu Jaccard, koeficientu podobnosti dveh podatkovnih nizov. Meri območje prekrivanja med napovedano vrednostjo (Area of Overlap) in pravo vrednostjo, ozirom unijo (Area of Union) (Slika 23). Izračuna se z enačbo (Rosebrock, 2022):

$$mIoU = \frac{TP}{TP+FP+FN} = \frac{Area\ of\ Overlap}{Area\ of\ Union}. \quad (37)$$

Če primerjamo $mIoU$ z določenim pragom t lahko določimo, ali je območje pravilno klasificirano ali ne.

$mIoU \geq t$... pravilno klasificirano

$mIoU < t$... napačno klasificirano



Slika 23: Primer prikaza pravilne in napovedane vrednosti (Rosebrock, 2022)

2.6 Metode ločevanja vodnih površin

Metode ločevanja vodnih površin potrebujemo za ustvarjanje podatkovnih regij, ki predstavljajo pravilne podatke v strojnem učenju. Pričakovani rezultat je binarna maska z binarno vrednostjo – je voda ali ni voda (vrednosti 1 in 0).

2.6.1 Razširjanje regij

Razširjanje regij je ena od metod za slikovno segmentacijo, ki temelji na izboru sosednjih pikslov in združenju le-teh, tako da tvorijo regijo. Osnovni princip ne zadostuje za vse vrste nalog, zato je bilo razvitih več verzij te metode. Nekatere od teh opisuje Pratt (Pratt, 2007). Definiran je le princip, ki sem

ga uporabil. Za izdelavo učne množice sem uporabil odprto-kodno implementacijo v sklopu (OpenLayers, 2021).

Glavni namen je izdelava rastrske slike z binarno vrednostjo iz vhodnega posnetka. Vhodni podatek je satelitski posnetek, ki se ga sestavi v barvni kompozit umetnih barv (B08, B04, B03), saj se s tem najboljše ločuje vodne površine. Prvi korak je izbira semena, oziroma izhodiščne točke na vhodnem posnetku. Sledi razširjanje regije toliko časa, dokler je razlika pod pragom ki se določi. Ta princip uporabi določanje sosednjih pikslov od semena glede na indeksirano barvno lestvico na posnetku. V primeru uporabe drugih vhodnih podatkov bi lahko sosedne določali glede na intenziteto pikslov ipd.

Prednost razširjanja regij je, da:

- zagotovi ostre robove regij,
- učinkovito ločuje regije,
- samostojno določanje semena in s tem postavljanje kriterija,
- omogoča izbiro več kriterijev hkrati in
- časovna učinkovitost, ker hkrati obravnava le sosednje piksele.

Slabosti so, da:

- je lokalna metoda, zato ne upošteva problema z globalnega vidika
- je občutljiva na šum.

Končni rezultat algoritma razširjanja regij je rastrska podoba, ki ima enako prostorsko ločljivost kot originalni posnetek. Ločena sta na dva razreda – vodna površina in ni vodna površina.

2.6.2 Normalizirani diferencialni indeks vode

Karakteristike svetlobne odbojnosti od vode se uporabljajo za označevanje površinskih voda. Večjo odbojnost imajo vidni spektralni kanali: rdeči, zeleni, modri (ang. red, green, blue - RGB). Najvišjo odbojnost ima modri kanal. Skoraj nič odbojnosti nimata bližnji infrardeči kanal (ang. Near Infrared - NIR) in kratkovalovni infrardeči kanal (ang. Short Wave Infrared – SWIR). Izračun normaliziranega diferencialnega indeksa vode se izračuna po enačbi:

$$NDWI = \frac{Green - NIR}{Green + NIR} \quad (38)$$

Poznamo tudi več modifikacij NDWI. Xu je (Xu, 2006) predlagal modificirano enačbo, kjer je namesto NIR spektralnega kanala uporabil SWIR, in jo poimenoval MNDWI. Novejše modifikacije enačbe so predlagali Madaresi Rod et al. v (Rad et al., 2021), ki od predhodnih različic da boljše rezultate in je primerna za različne scenarije. Bolje prepozna šum, ki ga povzroča temna vegetacija. Poimenovali so ga ANDWI.

Opredelil sem dve metodi, ki sta najpogosteje uporabljeni, vendar sem se v eksperimentalnem delu odločil za metodo razširjanja regij.

3 VIRI IN PREDOBDELAVA PODATKOV

3.1 Iskanje dogodkov in priprava geometrije

Predpriprava je zajemala izbiro ustrezne strojne in programske opreme. Uporabljena programska oprema se je sproti prilagajala. Vsebovala je:

- GMS-GIS 5.9.8 (Flycom Technologies d.o.o., 2021),
- Sentinel Hub (Sinergise d.o.o., 2021a),
- Pycharm 2021.2.4 (JetBrains d.o.o., 2021b), DataGrip 2022.2.4 (JetBrains d.o.o., 2021a),
- Python 3.6 (Tensorflow, SentinelHub.PY, Numpy),
- PostgreSQL 13.7 + PostGIS 3.1.0 in
- Wandb.ai.

Prva naloga je bila poiskati dogodke, na katerih bo potekalo učenje algoritmov. Dogodki se med seboj razlikujejo, s čimer sem želel doseči čim boljše učenje modelov, saj se s tem pridobi uspešnost na raznolikih poplavnih dogodkih. Velika omejitev je bila razpoložljivost podatkov, saj so vremenske razmere v obdobju poplav običajno oblačne. Pojavljajo se cirus oblaki, ki se lahko nehote spregledajo. Za nekatera območja, je zaradi boljše vidnosti, bil izbran kasnejši datum, kljub manjšemu obsegu poplave. Vsi dogodki so opazovani na treh kontinentih, in sicer v Evropi, Aziji in Severni Ameriki. Večina poplav se dogaja v jesenskem času, z izjemo poplave reke Mississipi, ki se je zgodila v spomladanskem času in poplave reke Ren, ki je poplavljala v zimskem času (na posnetkih so vidne površine pokrite s snegom).

Za iskanje dogodkov sem uporabil dva vira, in sicer Wikipedijo, kjer se sproti dodajajo, ažurirajo, dogodki najnevarnejših poplav (Wikipedia, 2021) in Copernicusovo službo za upravljanje v sili in kartiranje dogodkov (European Union, 2021).

Preglednica 4: Izbrani dogodki za vektorizacijo podatkov

Država	Ime	Vrsta poplave	Datum dogodka ¹	Datum satelitskega posnetka	Površina zajetega območja [km ²]
Italija	reka Sesia	rečna	3.10.2020	3.10.2020	172
Nemčija	reka Ren	rečna	1.2.2021	11.2.2021	211
Nigerija	reka Niger	rečna	15.9.2020	15.9.2020	319
Nigerija	reka Niger	rečna	31.8.2020	31.8.2020	82
Nigerija	reka Niger	rečna	12.9.2020	12.9.2020	1565
Italija	južna Sardinija	hitra (morje)	10.10.2018	11.10.2018	30
Romunija	reka Prut	rečna	26.6.2020	27.6.2020	46
Togo	reka Oti	rečna	5.10.2020	8.10.2020	371
ZDA	reka Mississippi	rečna	22.3.2019	22.3.2019	5088

¹ Poplave običajno trajajo več dni, zato datumi niso popolnoma referenčni in so odvisni tudi od poplavljenega kraja ob reki.

Za pregled dostopnosti podatkov (datum in vidnost posnetkov) sem uporabil program GMS-GIS. V programu sem ustvaril WMS sloj, ki je sestavljen v barvnem kompozitu umetnih barv. Vsebuje tri spektralne kanale Sentinel-2 B08, B04 in B03. Podatke za WMS sem pridobil na SentinelHub. Za pregled vodnih območij, kot so kartirane v nepoplavnem času, sem ustvaril sloj topografske karte HERE (HERE, 2021) (slika 24). Ob spreminjanju datuma in lokacije poplavnih dogodkov iz omenjenih virov sem zabeležil datum satelitskih posnetkov (preglednica 4). Za izbrane dogodke sem nato uporabil algoritem razširjanja regij za klasifikacijo vodnih območij do kopnega. Ročno sem izbiral piksele (semena) na vsakem posnetku in za vsakega določil prag, na podlagi katerega se je določilo območje regije. Določena regija je zapisana v rastrski obliki, katero se pretvori v vektorsko obliko, pri čemer so upoštevane geometrijske in topološke lastnosti originalnega podatka. Vektorska geometrija se nato shrani v tabelo v obliki WKT (Well-Known-Text) s pripadajočimi metapodatki (dogodek, datum ...).

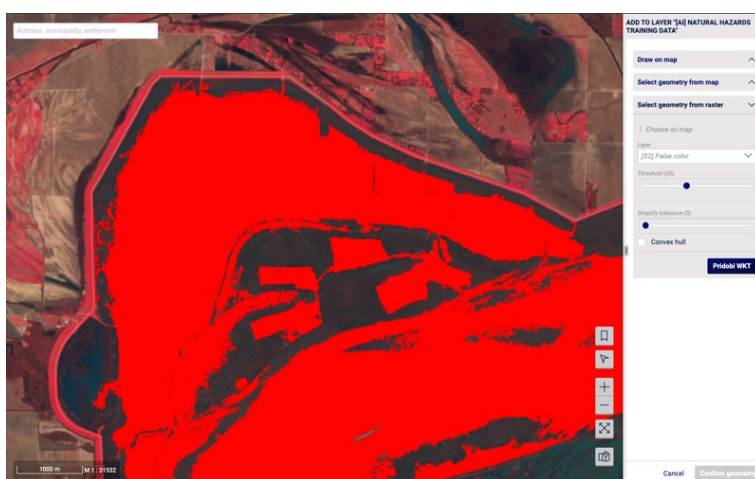
Slika 24 prikazuje topografsko karto, na kateri je razviden tok reke Mississippi. Na sliki 25 je v umetnih barvah (B08, B04, B03) prikazan optični satelitski posnetek, na katerem je viden začetek postopka razširjanja regij s pragom 5 (rdeča barva). Ker izbrano območje ne prekrije celotne poplave, sem povečal prag na 20, kot je prikazano na sliki 26. Prag sem v programu GMS-GIS določil na 8-bitni sliki, pri čemer sem uporabil vse tri kanale iz barvnega kompozita umetnih barv. Izbrano območje se vektorizira in se shrani kot vektor. Delo zajema učnih podatkov oziroma pravih območij za učenje je potekalo polavtomatsko.



Slika 24: Sloj topografske karte HERE za prikaz poteka reke



Slika 25: Izbrano območje (obarvano rdeče) s pragom 5



Slika 26: Izbrano območje (obarvano rdeče) s pragom 20

Po končanem zajemu geometrije z uporabo programskega jezika PostgreSQL + PostGIS združimo geometrije po dogodkih. Ker so novo nastali poligoni preveliki za procesiranje in samo prikazovanje, so bili razdeljeni v kvadratno mrežo manjših območij. S tem se je povečala hitrost procesiranja za pridobitev surovih podatkov in pravih vrednosti.

3.2 Sentinel Hub in nastavitve slojev

Sentinel Hub je platforma, ki jo je izdelalo podjetje Sinergise d.o.o. Zagotavlja tako surove kot procesirane satelitske podatke. Omogoča tudi poljubno sestavljanje spektralnih in analitičnih kanalov (predprocesirani z različnimi indeksi, na primer vegetacijski indeks) ter generiranje po meri novih analitičnih kanalov. Za različne potrebe se podatke lahko po želji spreminja po formatu ali vnaša druge prilagoditve. Platforma je namenjena strojnemu učenju, delu razvojnih inženirjev, analitikom idr. Ponuja več vrst podatkov, tako optičnih in radarskih ter drugih. Pridobljeni so iz več satelitov, kot so Sentinel, Landsat in drugih, ki ponujajo brezplačne in plačljive podatke. (Sinergise d.o.o., 2021b).

Prenos določenih satelitskih podatkov je brezplačen. V kolikor se potrebuje večja količina podatkov, se lahko le te pridobi z naročnino na storitev. (Sinergise d.o.o., 2020). Podatke je na brezplačen način mogoče pridobiti le kot posamezne satelitske posnetke. V mojem primeru je bila potrebna velika količina prenosa in procesiranja podatkov, zato ima v ta namen Sinergise pripravljeno posebno platformo na Sentinel Hub, kjer se nastavijo sloji z generiranjem JavaScript skripte, s katero se definirajo spektralni kanali. Prav tako se mora poenotiti format podatkov za pravilno delovanje učenja modela, torej se jih normalizira. Ker se obravnava več različic podatkovnih kombinacij, je posledično tudi za vsak sloj napisana posebna skripta.

Za ovrednotenje sem izbral štiri kombinacije podatkovnih struktur:

1. Sentinel-2 L1C (B02, B03, B04, B08, B11, B12 + CLP in dataMask)

Za prvo kombinacijo sem izbral šest spektralnih kanalov, verjetnost oblakov (CLP) in masko podatkov (dataMask). Za izbrane spektralne kanale sem se odločil, ker se s temi izračunavajo indeksi za prepoznavo vodnih površin. Opredelijo jih tudi v (Goffi et al., 2020).

2. Sentinel-2 L1C (vsi spektralni kanali + CLP in dataMask)
3. Sentinel-2 L2A (B02, B03, B04, B08, B11, B12 + CLD, SNW, CLP in dataMask)
4. Sentinel-2 L2A (vsi spektralni kanali + CLD, SNW, CLP in dataMask)

3.3 Programski del za prenos podatkov

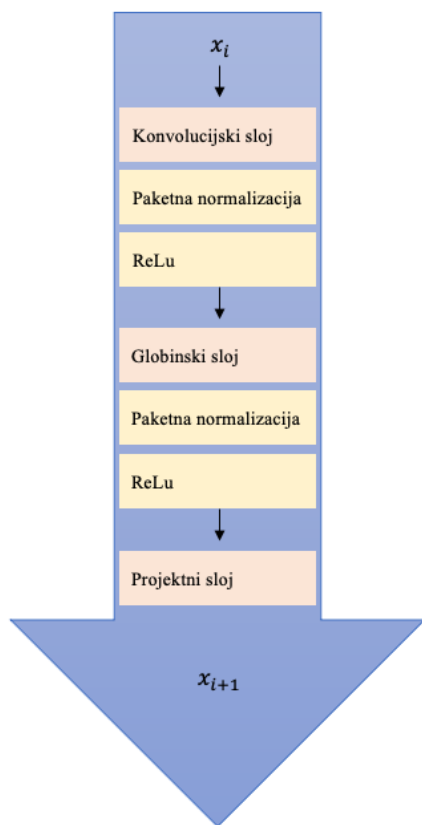
Za prenos podatkov sem izdelal programsko skripto v Python 3.6. Skripta je pripravljena na način, da z vnosom v ukazni vrstici poženemo ukaz za začetek obdelave podatkov. Ključnega pomena, da bo skripta delovala, je predhodna povezava na Sentinel Hub in povezava z bazo, kjer so shranjene tabele s podatki o dogodkih in WKT-geometriji. Najprej sem definiriral ime sloja na Sentinel Hubu, nato se povezal z podatki iz baze, kjer so dogodki shranjeni. Prevzame se ime dogodka, WKT-geometrijo dogodkov in datum satelitskih posnetkov, na katerih so dogodki vidni. Pred uporabo geometrije se naredi še transformacijo v najbolj ugodno cono UTM, saj sem podatke zajel v WebMercator, ki ni primeren za analize. V skripti je prav tako zapisano, da se izvede generiranje rastrskih binarnih mask iz zajete WKT-geometrije, ki opredeljuje poplavne in nepoplavne površine. Binarna maska je, z izjemo globine po dimenziji, enako velika kot satelitski posnetek. V primeru klasifikacije več tipov naravnih nesreč, bi piksel binarne maske imel številne možne vrednosti. Posnetki se po obdelavi shranijo v določen direktorij kot 16-bitno datoteko v zapisu TIFF. Shranijo se spektralni kanali kot tudi ostali kanali, ki so bili definirani na platformi Sentinel Hub (v prejšnjem poglavju navedene kombinacije). Povezljivost med satelitskimi posnetki in binarnimi maskami ohranimo z enotnim poimenovanjem datotek.

Prav tako je bila uporabljena tehnika bogatenja količine podatkov. Prva tehnika je prekrivanje posnetkov, ki znaša 400 m. Uporabljena tehnika je tudi naključna osvetlitev z delto 0,2 %. Sledi še naključna sprememba kontrasta, izbrana naključno iz intervala [0,07 in 0,08]. Zadnja od uporabljenih tehnik bogatenja količine podatkov je naključna rotacija. Izbral sem še naključne posnetke, ki se trikrat rotirajo za 90 stopinj. Rotacija se je, enako kot za satelitske posnetke, uporabila tudi za binarne maske.

4 ARHITEKTURA ZGRAJENIH MODELOV IN ŠTEVILO PARAMETROV

4.1 MobileNetV2

Zgrajen model MobileNetV2 je sestavljen iz 16 blokov in je na sliki 28, prikazan v delu imenovanem »model: Functional« (od 16. so prikazani samo uporabljeni bloki). Ob podrobni razdelitvi modela sem ugotovil, da se dimenzije posnetkov, ki so potrebne za vhodne podatke, nahajajo na blokih 1, 3, 6, 13 in 16. Tako je dimenzija vhoda 256×256 , kjer se za vsak blok potrebuje razpolovljena vrednost prejšnjega bloka. Navede se v grafu pod vrednostjo »output«, in sicer [(None, 128, 128, 96), (None, 64, 64, 144), (None, 32, 32, 192), (None, 16, 16, 576), (None, 8, 8, 329)], kjer so vrednosti 128, 64, 32, 16 in 8 velikosti dimenzij. Te so izbrane, kajti vhodni podatek je dimenzije 256×256 . Globina se pri tem spreminja zaradi operacij v skritih slojih bloka. Osnovna sestava bloka je prikazana na sliki 27.



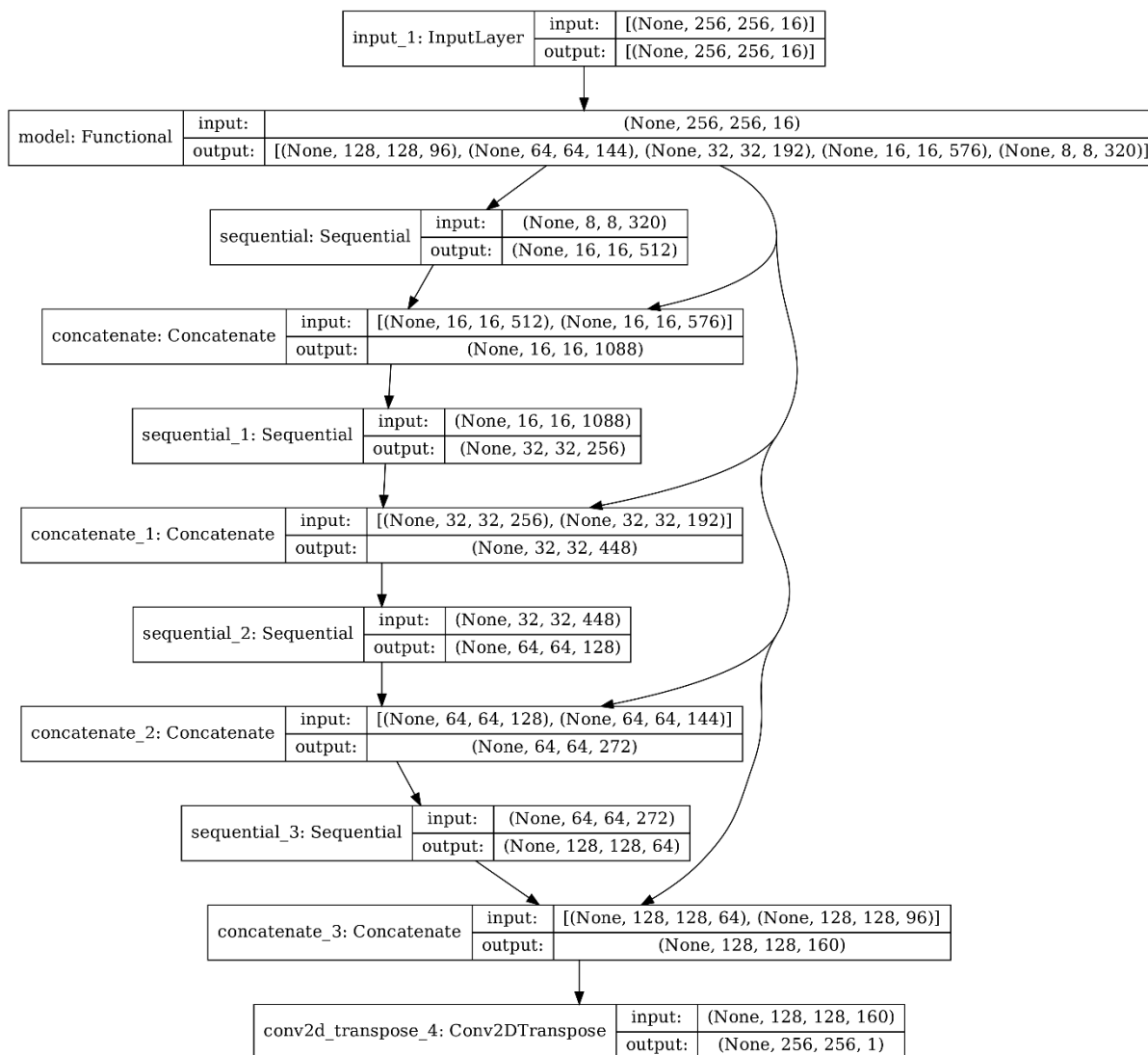
Slika 27: Osnovna sestava bloka - MobileNetV2

Razširil sem zgrajen blok, opravil paketno normalizacijo in uporabil aktivacijsko funkcijo ReLU. Izhodni podatek tega bloka je vhodni podatek za naslednji blok. V omenjenih petih blokih je dodan tudi sloj dodajanja ene dimenzije roba (padding).

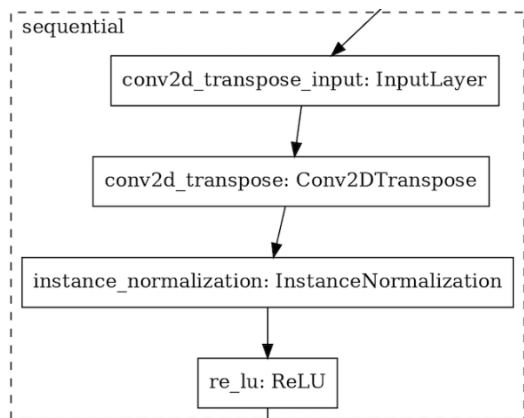
Izhodne podatke arhitekture modela MobileNetV2 sem uporabil za nadzorčenje, saj se potrebuje rezultat v enakih dimenzijah kot vhodni podatki. Uporabil sem zaporedne (ang. sequential) bloke (slika 29) in združevalne (ang. concatenate) sloje (slika 28). Zaporedni blok je sestavljen iz štirih slojev. Vhodni podatek izvira iz modela ali združevalnega sloja. Za nadzorčenje sem uporabil konvolucijski

sloj za transponiranje, nato podatke normaliziral z uporabo normalizacije primerka ter uporabil aktivacijsko funkcijo ReLU. Zadnji sloj na grafu na sliki 28 predstavlja le konvolucijski sloj za transponiranje, ki uporablja sigmoidno aktivacijsko funkcijo, da podatke nadzorči na dimenzijo enako vhodnim podatkom ob globini 1.

Za konec sem še izračunal število parametrov in s tem pridobil informacijo o moči modela za napovedovanje vrednosti. Izračun je povzet v preglednici 5 in vrne 6,47 milijon učnih parametrov.



Slika 28: Graf zgrajenega modela MobileNetV2 po blokih



Slika 29: Graf bloka "sequential"

Preglednica 5: Izračun števila parametrov po blokih - MobileNetV2

Sloj (tip)	Izhodna oblika	Št. Parametrov
input 1 (InputLayer)	(None, 256, 256, 16)	0
model (Functional)	[(None, 128, 128, 96), (None, 64, 64, 144), (None, 32, 32, 192), (None, 16, 16, 576), (None, 8, 8, 320)]	1845728
sequential (Sequential)	(None, 16, 16, 512)	1475584
concatenate (Concatenate)	(None, 16, 16, 1088)	0
sequential_1 (Sequential)	(None, 32, 32, 256)	2507264
concatenate_1 (Concatenate)	(None, 32, 32, 448)	0
sequential_2 (Sequential)	(None, 64, 64, 128)	516352
concatenate_2 (Concatenate)	(None, 64, 64, 272)	0
sequential_3 (Sequential)	(None, 128, 128, 64)	156800
concatenate_3 (Concatenate)	(None, 128, 128, 160)	0
conv2d_transpose_4 (Conv2DTrans)	(None, 256, 256, 1)	1441
Seštevek parametrov: 6,503,169		
Učljivi parametri: 6,472,257		
Neučljivi parametri: 30,912		

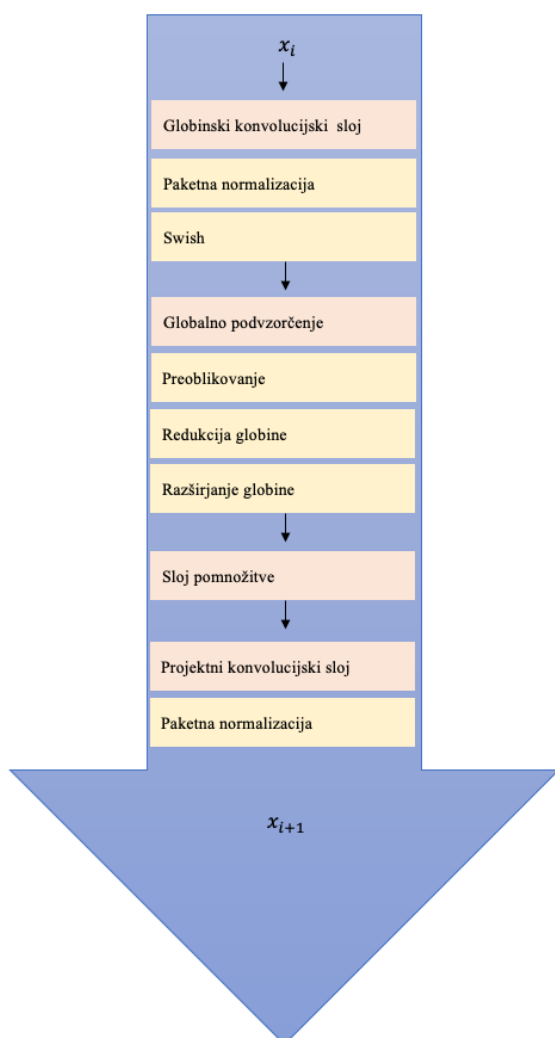
4.2 EfficientNet-B0

Model EfficientNet-B0 sestavlja 7 blokov, kjer se vsak razdeli na več pod-blokov. Podobno kot pri prejšnjem modelu je tudi tukaj dimenzija vhodnih podatkov 256 x 256. Ker je za vsak blok potrebna razpolovljena vrednost, po metodi podvzorčenja sem pregledal, kateri bloki modela EfficientNet-B0 ustrezajo tem dimenzijam. To so bloki 1a, 2b, 3b, 5c in 7a. Osnovna sestava bloka je prikazana na sliki 30.

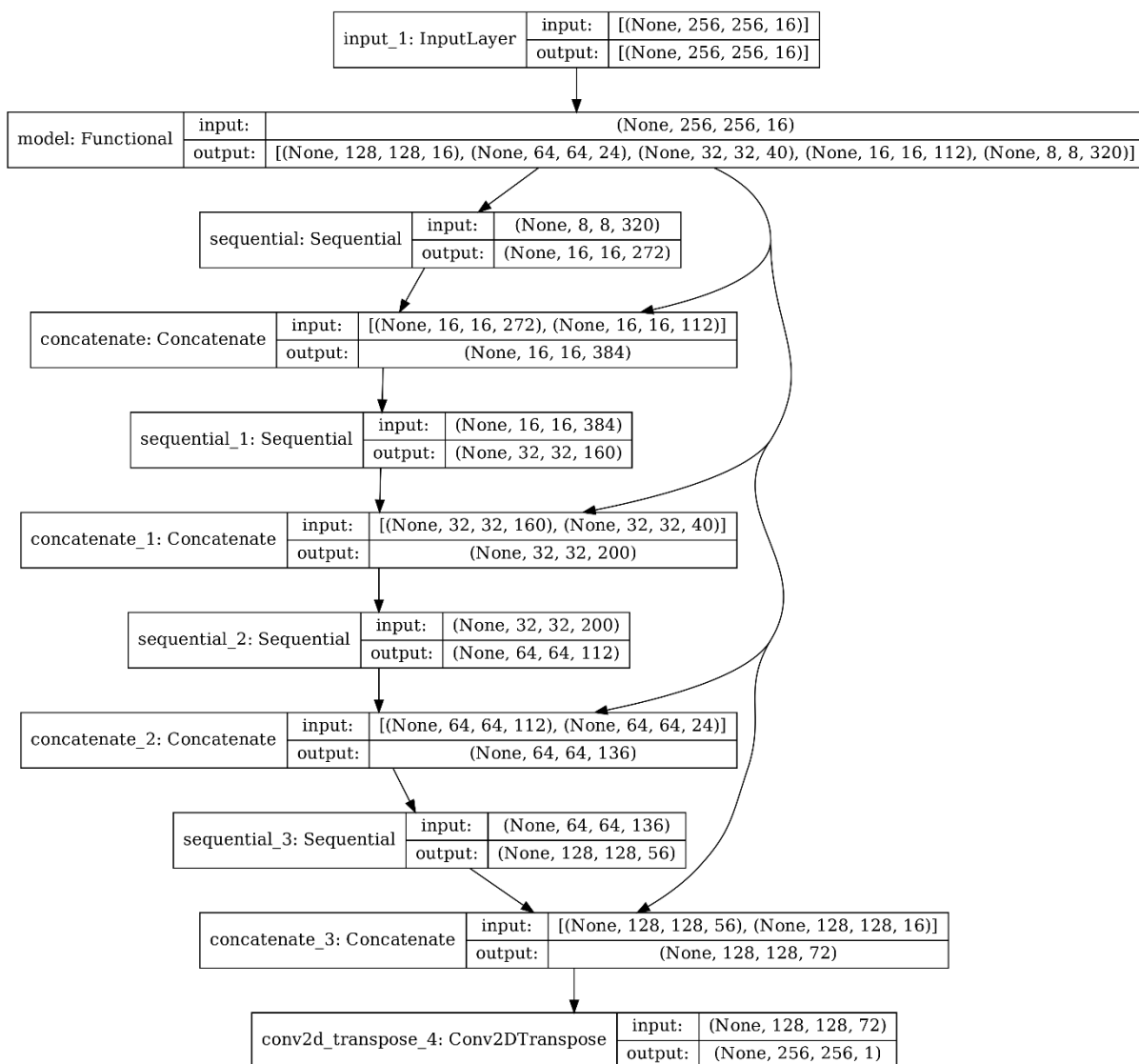
V določenih blokih se uporabijo tudi sloji dodajanja ene dimenzije (padding), izpadnega sloja (ang. dropout) in dodajanj (ang. add).

S tem se pridobi definiran model EfficientNet-B0. Postopek nadvzorčenja je tukaj enak kot v modelu MobileNetV2 – enake dimenzije, vendar z drugimi globinami. Za zadnji konvolucijski sloj za transponiranje je bila uporabljena sigmoidna aktivacijska funkcija. Grafični prikaz je dodan na sliki 31.

Na enak način se izračuna še število parametrov, ki so povzeti v preglednici 6. Vsota števila učnih parametrov je 5,21 milijona.



Slika 30: Osnovna sestava bloka - EfficientNet-B0



Slika 31: Graf zgrajenega modela EfficientNet-B0 po blokih

Preglednica 6: Izračun števila parametrov po blokih - EfficientNet-B0

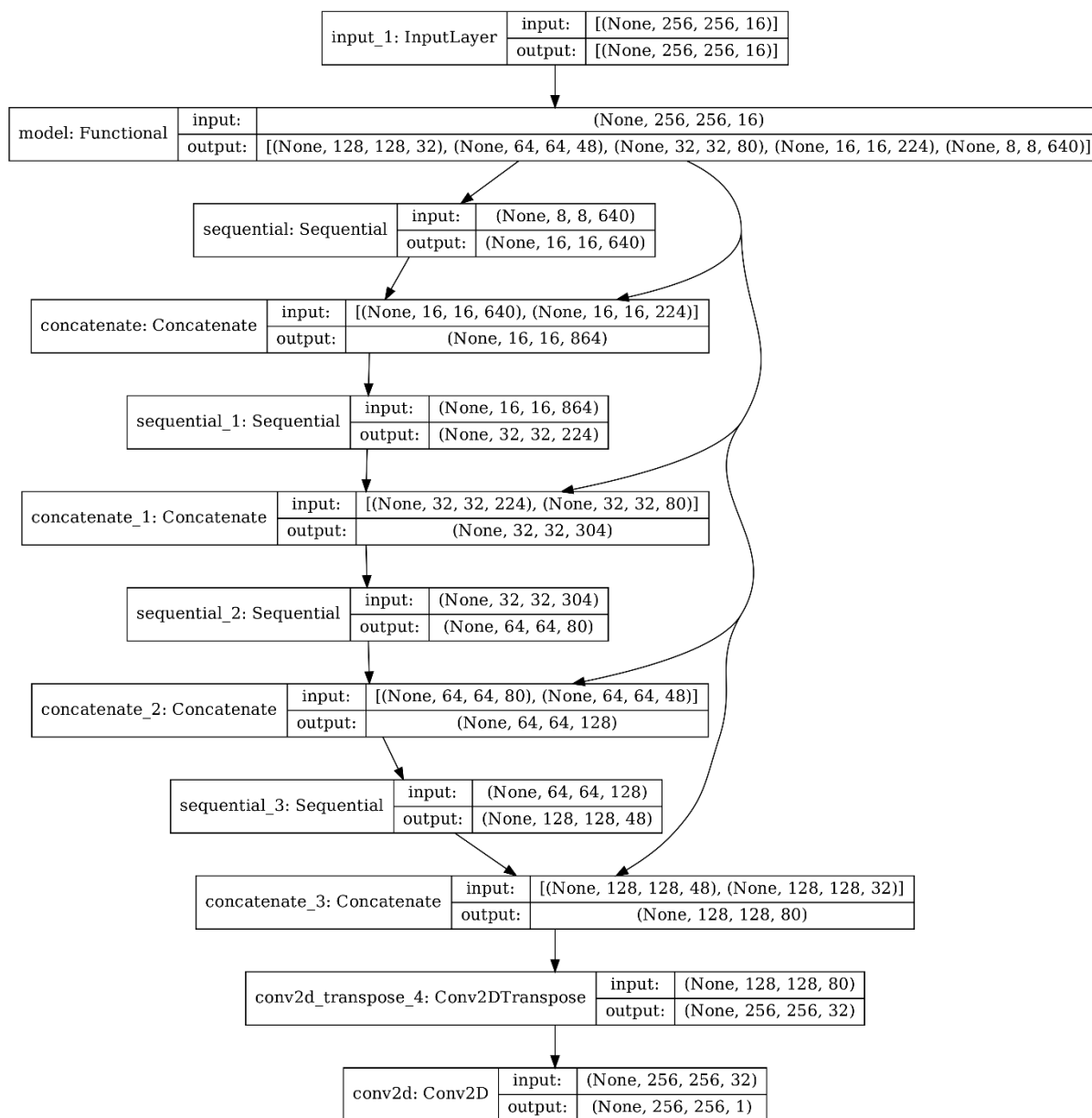
Sloj (tip)	Izhodna oblika	Št. Parametrov
input_1 (InputLayer)	(None, 256, 256, 16)	0
model (Functional)	[(None, 128, 128, 16), (None, 64, 64, 24), (None, 32, 32, 40), (None, 16, 16, 112), (None, 8, 8, 320)]	3638621
sequential (Sequential)	(None, 16, 16, 272)	783904
concatenate (Concatenate)	(None, 16, 16, 384)	0
sequential_1 (Sequential)	(None, 32, 32, 160)	553280
concatenate_1 (Concatenate)	(None, 32, 32, 200)	0
sequential_2 (Sequential)	(None, 64, 64, 112)	201824
concatenate_2 (Concatenate)	(None, 64, 64, 136)	0
sequential_3 (Sequential)	(None, 128, 128, 56)	68656
concatenate_3 (Concatenate)	(None, 128, 128, 72)	0
conv2d_transpose_4 (Conv2DTrans)	(None, 256, 256, 1)	649
Seštevek parametrov: 5,246,934		
Učljivi parametri: 5,207,445		
Neučljivi parametri: 39,489		

4.3 EfficientNet-B7

Model EfficientNet-B7 deluje na enak način kot B0, zajema 7 blokov, vendar ima zraven tega veliko več razvejitenih blokov. Osnovna zgradba bloka je enaka kot pri B0. Tukaj sem uporabil bloke 1d, 2g, 3g, 5j in 7d, da postopku ustrezajo dimenzije. Če se primerja izhodne podatke tega modela z izhodnimi podatki prejšnjega modela, se ugotovi, da je razlika le v globini.

Postopek nadvzorčenja je enak kot pri prejšnjih dveh modelih z izjemo predzadnjega konvolucijskega sloja za transponiranje, saj je uporabljena aktivacijska funkcija ReLU. Prav tako je dodan še en sloj kot zadnji, ki uporabi sigmoidno aktivacijsko funkcijo. Grafični prikaz osnovnih blokov je prikazan na Slika 32.

Enako kot v prejšnjih modelih, je tudi tukaj bil izveden postopek izračuna števila parametrov, z rezultatom 67,88 milijona. Povzetek izračuna je prikazan v preglednici 7.



Slika 32: Graf zgrajenega modela EfficientNet-B7 po blokih

Preglednica 7: Izračun števila parametrov po blokih - EfficientNet-B7

Sloj (tip)	Izhodna oblika	Št. Parametrov
input_1 (InputLayer)	(None, 256, 256, 16)	0
model (Functional)	[(None, 128, 128, 32), (None, 64, 64, 48), (None, 32, 32, 80), (None, 16, 16, 224), (None, 8, 8, 640)]	62456561
sequential (Sequential)	(None, 16, 16, 640)	3687680
concatenate (Concatenate)	(None, 16, 16, 864)	0
sequential_1 (Sequential)	(None, 32, 32, 224)	1742272
concatenate_1 (Concatenate)	(None, 32, 32, 304)	0
sequential_2 (Sequential)	(None, 64, 64, 80)	219040
concatenate_2 (Concatenate)	(None, 64, 64, 128)	0
sequential_3 (Sequential)	(None, 128, 128, 48)	55392
concatenate_3 (Concatenate)	(None, 128, 128, 80)	0
conv2d_transpose_4 (Conv2DTrans)	(None, 256, 256, 32)	23072
conv2d (Conv2D)	(None, 256, 256, 1)	33
Seštevek parametrov: 68,184,050		
Učljivi parametri: 67,878,417		
Neučljivi parametri: 305,633		

5 REZULTATI UČENJA MODELOV IN ANALIZA

V eksperimentalnem delu magistrskega dela obravnavam tri modele CNN, in sicer MobileNetV2, EfficientNet-B0 in EfficientNet-B7. Med seboj se razlikujejo predvsem po kompleksnosti in številu parametrov (preglednica 8). Za primerjavo je v spodnji preglednici navedeno, koliko parametrov ima posamezen model in čas, ki ga potrebuje za učenje. Upoštevani so hiperparametri iz prvega dela testiranja. Uporabil sem grafično procesno enoto (GPU) *NVIDIA GeForce GTX 1080 Ti* s pomnilnikom 11 GB.

Preglednica 8: Prikaz števila parametrov in čas učenja na enem koraku (Keras, n.d.)

Model	Število parametrov (v milijonih)	Čas za učenja modela (minute)
MobileNetV2	6,47	71
EfficientNet-B0	5,21	79
EfficientNet-B7	67,88	200

Za učenje sem uporabil 90 % podatkov in 10 % za validacijo. Razdelitev podatkov je bila naključna.

Za določene hiperparametre sem se odločil uporabiti ugotovitve iz predhodnih raziskav in testirati le optimizator, kriterijsko funkcijo in stopnjo učenja. Normalizacija je v vseh testih enaka, in sicer normalizacija primerka, kjer sem upošteval ugotovitve iz raziskave (Ulyanov et al., 2016), v kateri trdijo, da z uporabo normalizacije primerka, namesto paketne normalizacije, izboljšajo učinkovitost učenja. Za zmanjševanje stopnje učenja (LRD) sem uporabil eksponentno zmanjševanje stopnje učenja, katerega v raziskavi *Razpored eksponentnih stopenj učenja za globoko učenje* (Z. Li & Arora, 2019) opredelijo za uspešnega. Izbrana velikost paketa (ang. batch size) je 8 in izbrano število epoh je 30. V vseh delih testiranja sem uporabljal omenjene nespremenljive hiperparametre. Razen v tretjem delu sem preizkusil povečati število epoh.

Za spremljanje metrik modelov in sprotne analize sem uporabil spletno platformo wandb.ai, ki omogoča sprotno nalaganje metrik. Nanjo se shranijo vsi podatki, ki jih definiramo, to pomeni vse sprotne metrike, ki jih uporabimo za izris spreminjanja metrik. Naložimo lahko tudi rezultate in izvedemo sprotne poročila in analize.

Delo je potekalo v treh korakih:

1. Testiranje izbranih hiperparametrov

Glede na predelana gradiva sem pri vsakem hiperparametru izbral po dve optimalni možnosti, saj sem s tem prihranil na času in količini testiranja. Prav tako sem se omejil na zmogljivost razpoložljive grafične procesne enote (GPU). V preglednici 9 so našteje vrednosti v obravnavi.

Preglednica 9: Hiperparametri v prvem delu testiranj

Hiperparameter	Vrednosti 1	Vrednost 2
Optimizator	Adam	RMSprop
Funkcija izgube	Binarna prečna entropija (BCE)	Dice
Stopnja učenja	0,001	0,0001

Za vsak izbran model sem testiral vse možne kombinacije hiperparametrov iz preglednice 8, tako da je vseh testov v prvem delu 24. Ustrezna kombinacija hiperparametrov se pri različnih modelih razlikuje, saj se ti razlikujejo po arhitekturi. Rezultat sem predstavil kot najboljše kombinacije hiperparametrov na posamezen model.

2. Test spreminjanja podatkovne strukture

Uporabil sem tri modele z najuspešnejšimi finimi nastavitvami iz prvega dela testiranj. V drugem delu testiranj sem spreminjal podatkovno strukturo vhodnih podatkov. Spreminjal sem produkt Sentinel-2, število spektralnih kanalov in število ostalih kanalov. Vrednosti so navedene v preglednici 10. Število kanalov se zaradi razpoložljivosti podatkov razlikuje glede na produkt (razpoložljivost v preglednici 2).

Preglednica 10: Spremembe podatkov v drugem delu testiranj

Spremenljivka (globina)	Vrednosti 1	Vrednost 2
Število spektralnih kanalov + številko ostalih kanalov za produkt L1C	6+2	13+2
Število spektralnih kanalov + številko ostalih kanalov za produkt L2A	6+4	12+4

Za vsak izbran model sem testiral vse možne kombinacije podatkovne strukture iz preglednice 10, tako da je število testov v drugem delu 12. Modeli s podatki L2A so po pričakovanju dali rezultate z najvišjo kakovostjo, kjer je upoštevana korekcija atmosferskega vpliva in dodatna dva kanala, ki vsebujeta natančnejšo napoved oblakov in snežnih površin. S tem se zmanjša vpliv prekomernega prileganja, v kolikor je prisoten. Rezultat drugega dela testiranj so trije najuspešnejši modeli.

3. Test večanja podatkov in števila epoh

V tretjem delu testiranj sem uporabil najuspešnejši model iz drugega dela testiranj. Spreminjal sem količino podatkov, torej večje prekrivanje podatkov in spremembo števila epoh v učenju modela. Pričakovani rezultat je boljša natančnost modela, z možnostjo prekomernega prileganja.

Za ocenjevanje uspešnosti modelov sem uporabil metriko AUC, ki najbolje predstavi povprečno natančnost modela. Za najuspešnejše modele po AUC sem izrisal še graf vrednosti izgube, iz katere ugotovimo, ali se model prekomerno prilega.

5.1 Rezultat testiranja hiperparametrov po modelu

Cilj prvega dela testiranja je bil izbrati najboljše parametre za posamezen model. Metrike, navedene v spodnjih tabelah, so izračunane iz validacijske množice. Imena v tabelah so avtomatsko generirana na spletni strani wandb.ai.

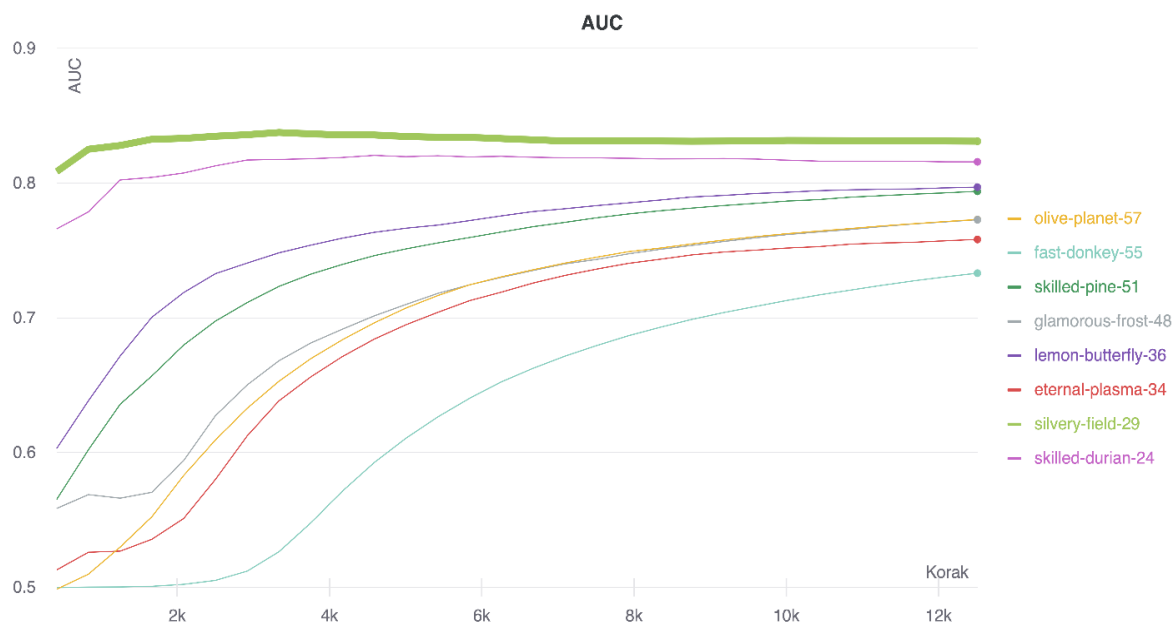
Vsi modeli v prvem delu testiranja imajo 12510 globalnih korakov učenja.

1. MobileNetV2

Iz rezultatov prvega modela iz preglednice 11 je razvidno, da razlika v metriki AUC variira za kar 10 odstotkov. Kot najboljši se je izkazal silvery-field-29, ki vsebuje kot optimizator hiperparametre rmsprop, dice kot kriterijsko funkcijo, ter vrednost 0.001 za stopnjo učenja (poudarjen v preglednici 11). Edini primerljivi model je skilled-durian-24, ki je sicer po AUC za 1,5 % manj zanesljiv, vendar z manjšo vrednostjo izgube. Slika 33 prikazuje vse teste tega dela za model MobileNetV2, kjer so razvidne velike razlike v AUC med testi. Zelena odebeljena krivulja predstavlja najuspešnejšo kombinacijo parametrov. Po obliki krivulje je zaznati, da se model hitro nauči – do približno 4000. koraka; nakar izboljšav ni več opaziti.

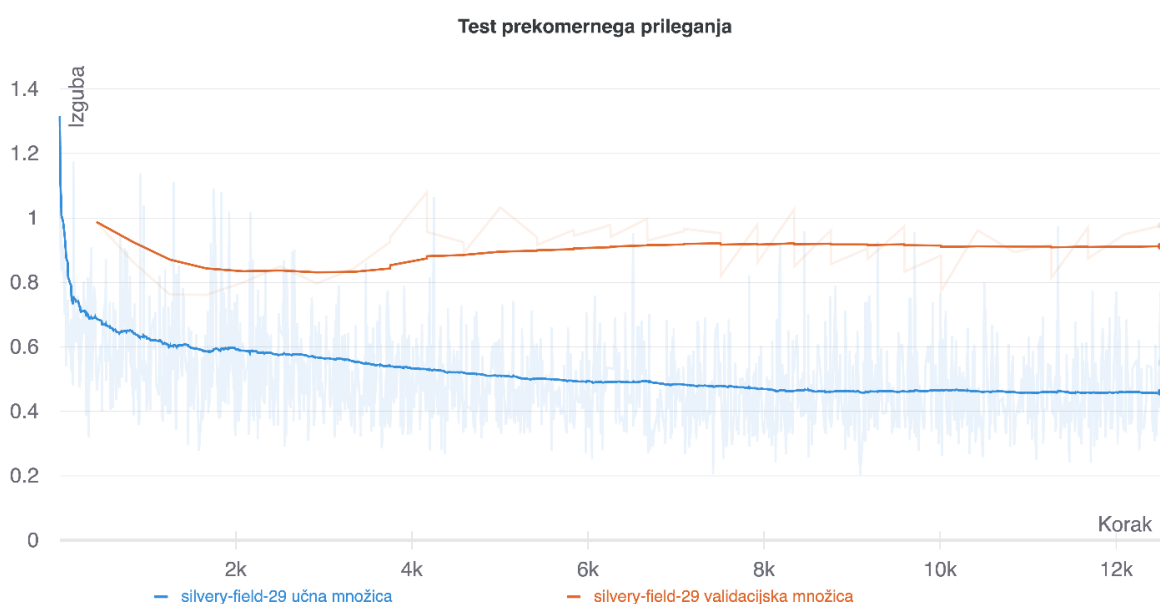
Preglednica 11: Rezultati prvega dela testiranja - MobileNetV2

Ime eksperimenta	Optimizator	Kriterijska funkcija	Stopnja učenja	AUC [%]
skilled-durian-24	rmsprop	bce	0,001	81,6
silvery-field-29	rmsprop	dice	0,001	83,1
eternal-plasma-34	rmsprop	bce	0,0001	75,8
lemon-butterfly-36	rmsprop	dice	0,0001	79,7
glamorous-frost-48	adam	bce	0,001	77,3
skilled-pine-51	adam	dice	0,001	79,4
fast-donkey-55	adam	bce	0,0001	73,3
olive-planet-57	adam	dice	0,0001	77,3



Slika 33: Grafični prikaz metrike AUC po korakih učenja v prvem delu testiranj - MobileNetV2

Slika 34 prikazuje potek vrednosti izgube, glede na korak učenja modela za učno in validacijsko množico. Uporabil sem glajenje krivulje (poudarjeno), saj iz originalne (nekoliko transparentno) krivulje ni mogoče dobro presoditi naraščanja ali padanja. Model se na stopnji prenehanja padca ali naraščanja ne izboljšuje. Vzrok je lahko premajhna učna množica, ki pa se z težavo poveča, ker zavzame 90 % vseh podatkov. Problem je rešljiv z izboljšanjem podatkov, kar navajam v drugem delu testiranja.



Slika 34: Potek vrednosti izgube za najboljšo kombinacijo hiperparametrov silvery-field-29 (MobileNetV2)

2. EfficientNet-B0

Drugi model, EfficientNet-B0, se je izkazal za nekoliko manj zanesljivega kot prvi MobileNetV2. Uporabljeni hiperparametri so v analizi enaki, vendar, v primerjavi z MobileNetV2, po uspešnosti kombinacije hiperparametrov obratni. V preglednici 12 se zazna, da najvišjo vrednost AUC, 80,6 %, doseže honest-water-56, ki vsebuje kot optimizator hiperparametre adam, bce kot kriterijsko funkcijo in vrednost 0.0001 za stopnjo učenja. Razlika v rezultatih med najuspešnejšim in najslabšim testom je le 2,5 % in drugim najboljšim, dry-down-58, pa le za 0,2 %, pri spremembi kriterijske funkcije. Iz Slika 35 ugotovimo, da ima honest-water-56 najustrežnejši potek krivulje, saj se na začetku postopno uči in ne pride do padca AUC. Prav tako kot v prvem modelu se na določeni točki ustavi izboljševanje modela.

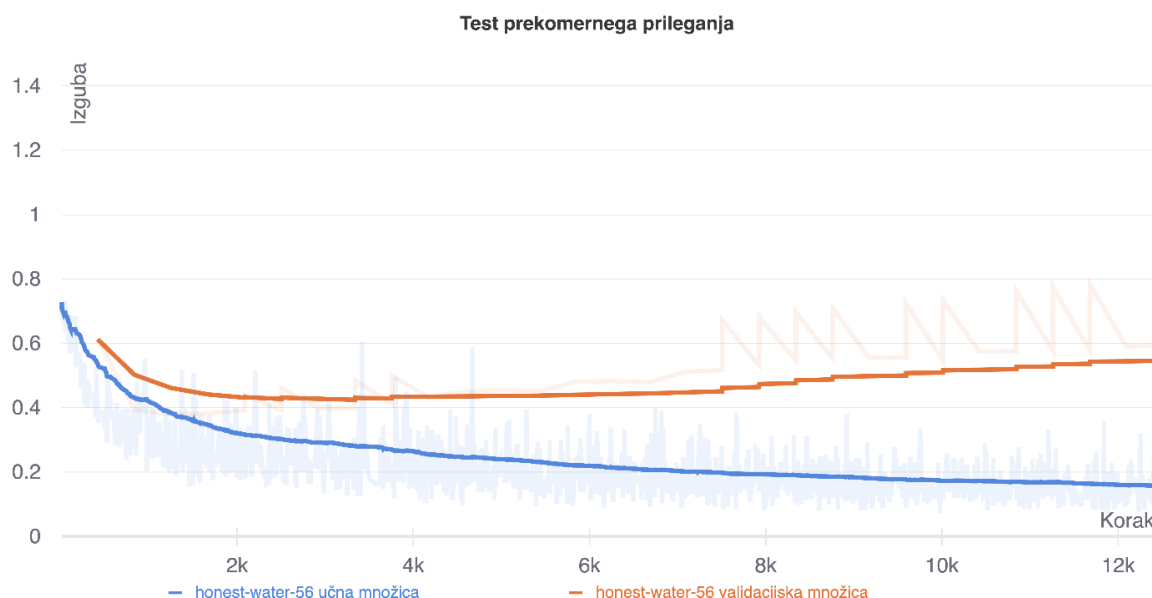
Preglednica 12: Rezultati prvega dela testiranj – EfficientNet-B0

Ime eksperimenta	Optimizator	Kriterijska funkcija	Stopnja učenja	AUC [%]
glad-sound-25	rmsprop	bce	0,001	79,4
effortless-flower-30	rmsprop	dice	0,001	79,5
effortless-jazz-35	rmsprop	bce	0.0001	78,7
golden-meadow-38	rmsprop	dice	0,0001	79,3
grateful-pond-50	adam	bce	0,001	78,1
charmed-haze-54	adam	dice	0,001	78,2
honest-water-56	adam	bce	0,0001	80,6
dry-dawn-58	adam	dice	0,0001	80,4



Slika 35: Grafični prikaz metrike AUC po korakih učenja v prvem delu testiranj - EfficientNet-B0

Na Slika 36, kjer je krivulja glajena, se razbere, da je približno na koraku 7500 začetek dvigovanja krivulje validacijske množice, medtem ko krivulja učne množice pada. Ta pojav imenujemo prekomerno prilaganje (ang. overfitting), kar pomeni, da se model že uči napak. Razlog za to je lahko, da je model preveč kompleksen ali je količina podatkov premajhna. Z problemom se soočim v drugem delu testiranj.



Slika 36: Potek vrednosti izgube za najboljšo kombinacijo hiperparametrov honest-water-56 (EfficientNet-B0)

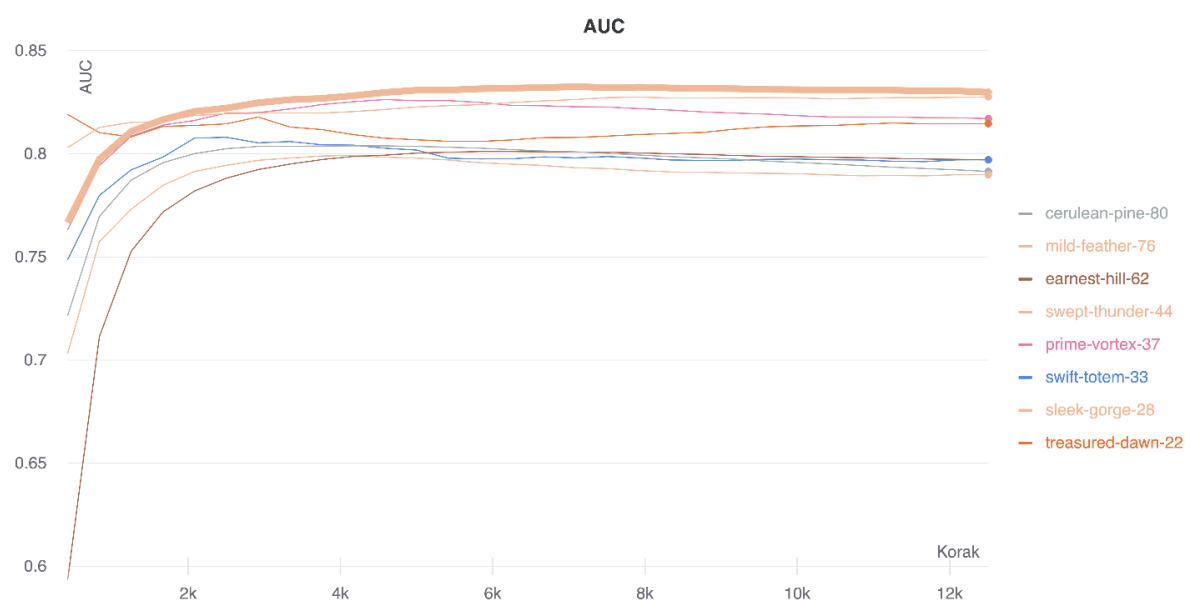
3. EfficientNet-B7

Zadnji model, EfficientNet-B7, je najkompleksnejši za učenje, ker potrebuje približno 12-krat več parametrov kot drugi model EfficientNet-B0 in je pri učenju počasnejši (odvisno od podatkov, omenjena primerjava se navezuje na podatke iz preglednice 7). Najboljši in najslabši iz preglednice 13 se po metriki AUC razlikujeta za 4 %. Pri tem ima najvišjo vrednost 83 % sleek-gorge-28, s hiperparametri rmsprop kot optimizator, dice kriterijsko funkcijo in stopnjo učenja 0,0001. Iz preglednice 15, kot tudi iz slike 37 je zaznati, da se dva najuspešnejša eksperimenta v metriki AUC razlikujeta samo za 0,2 %.

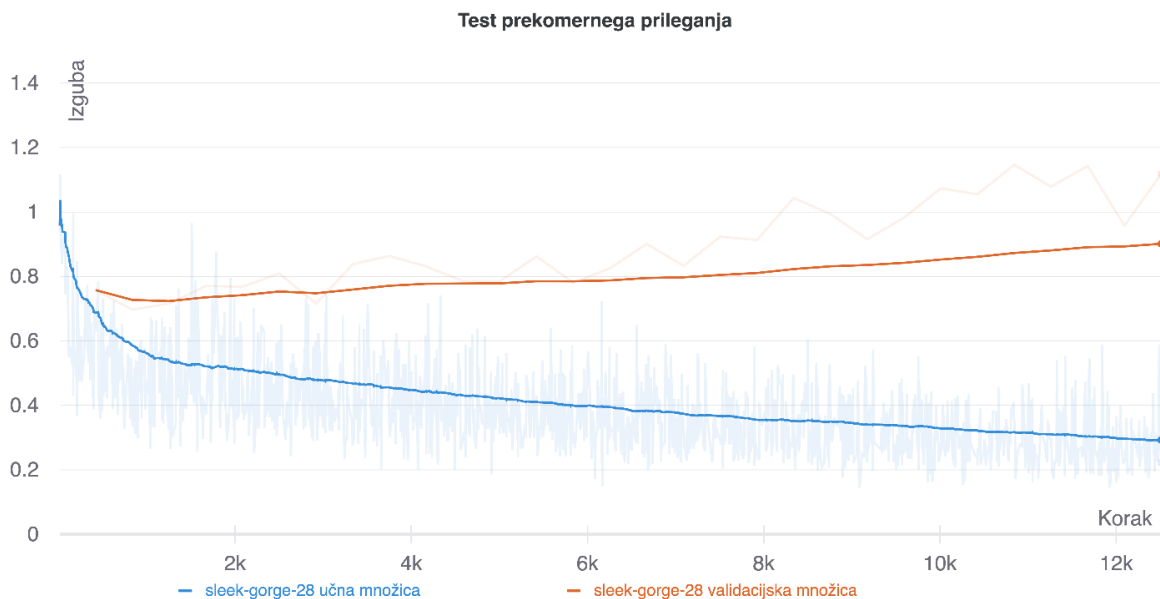
Preglednica 13: Rezultati prvega dela testiranj – EfficientNet-B7

Ime eksperimenta	Optimizator	Kriterijska funkcija	Stopnja učenja	AUC [%]
treasured-dawn-22	rmsprop	bce	0,001	81,5
mild-feather-76	rmsprop	dice	0,001	82,8
swept-thunder-44	rmsprop	bce	0,0001	79,0
sleek-gorge-28	rmsprop	dice	0,0001	83,0
swift-totem-33	adam	bce	0,001	79,7
prime-vortex-37	adam	dice	0,001	81,7
earnest-hill-62	adam	bce	0,0001	79,7
cerulean-pine-80	adam	dice	0,0001	79,1

Podobno kot v prejšnjem modelu opazimo na sliki 38, kot pojav prekomernega prileganja, vendar v tem primeru je nekoliko večji. Prav tako na samem začetku opazimo naraščanje vrednosti izgube. Vzrok tiči, kot sem omenil, v kompleksnejšem modelu za učenje – za kar se potrebuje veliko večja podatkovna množica. V drugem delu testiranj skušam kar se da optimizirati model in odpraviti prekomerno prileganje.



Slika 37: Grafični prikaz metrike AUC po korakih učenja v prvem delu testiranj – EfficientNet-B7



Slika 38: Potek vrednosti izgube za najboljšo kombinacijo hiperparametrov sleek-gorge-28 (EfficientNet-B7)

5.2 Rezultat testiranja sprememb podatkovne strukture po modelu

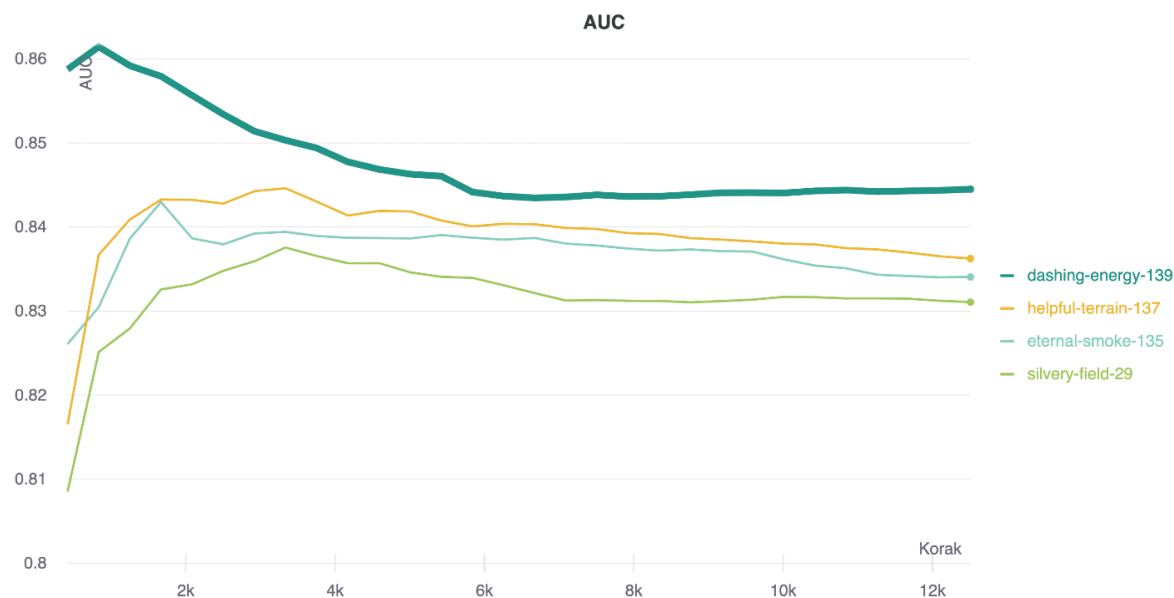
V drugem delu sem testiral, ali sprememba produkta Sentinel-2 in različna globina posnetkov (količina spektralnih in ostalih kanalov) izboljšata rezultat. V poglavju, Sentinel Hub in nastavitvev slojev, sem navedel kanale, ki so v posamezni globini. Glavni namen spremembe podatkov je izboljšanje metrike AUC in zmanjšanje prekomernega prekrivanja.

1. MobileNetV2

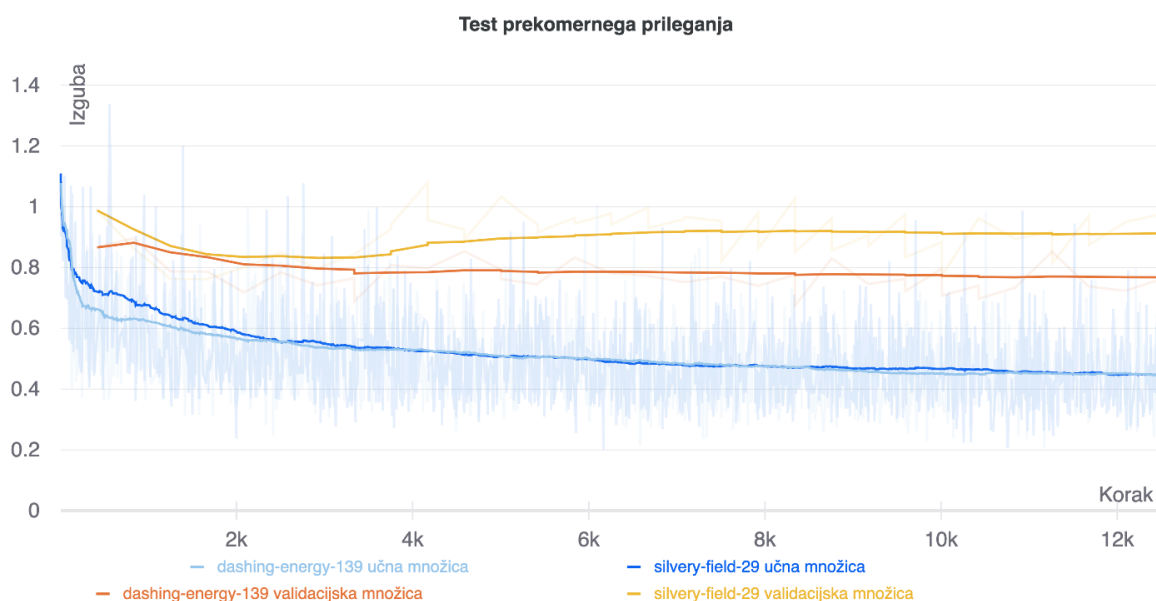
Iz preglednice 14 in slike 39 je razbrati, da se vrednosti izboljšajo, v primeru modela MobileNetV2 je izboljšava AUC za kar 1,4 %, pri spremembi S2 produkta iz L1C v L2A in podvojitvijo globine. Izboljša se tudi vrednost izgube za 0,12, kot je prikazano na sliki 40.

Preglednica 14: Rezultati drugega dela testiranj - MobileNetV2

Ime eksperimenta	S2 produkt	Globina	Izguba	AUC [%]
silvery-field-29	L1C	8	0,849	83,1
eternal-smoke-135	L1C	15	0,792	83,4
helpful-terrain-137	L2A	10	0,941	83,6
dashing-energy-139	L2A	16	0,729	84,5



Slika 39: Grafični prikaz metrike AUC po korakih učenja v drugem delu testiranj - MobileNetV2



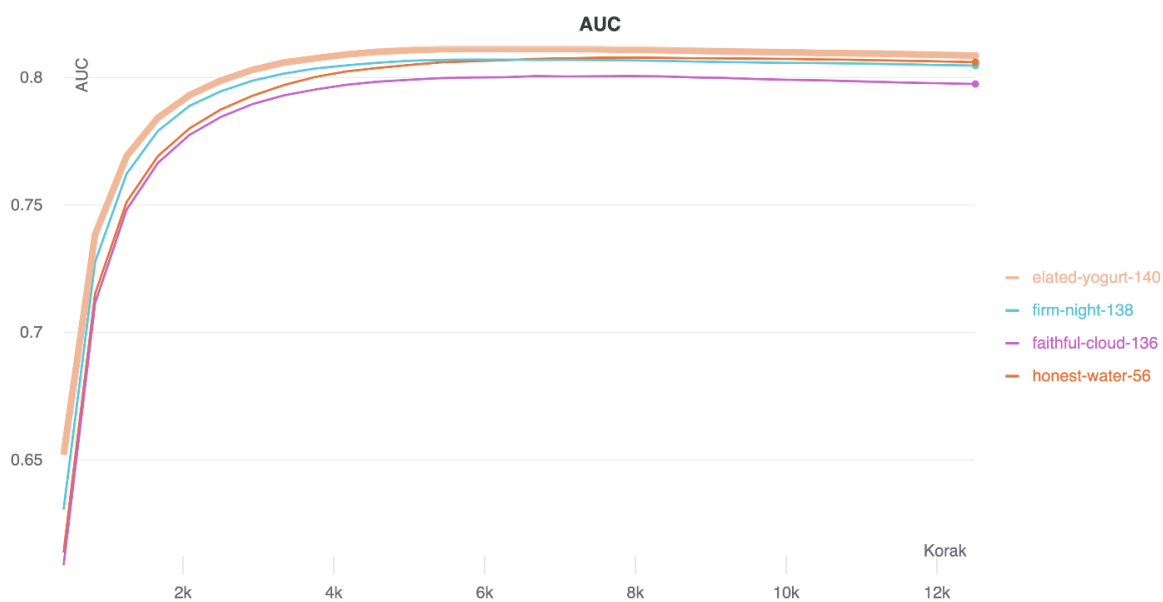
Slika 40: Potek vrednosti izgube pri dveh najboljših testih - MobileNetV2

2. EfficientNet-B0

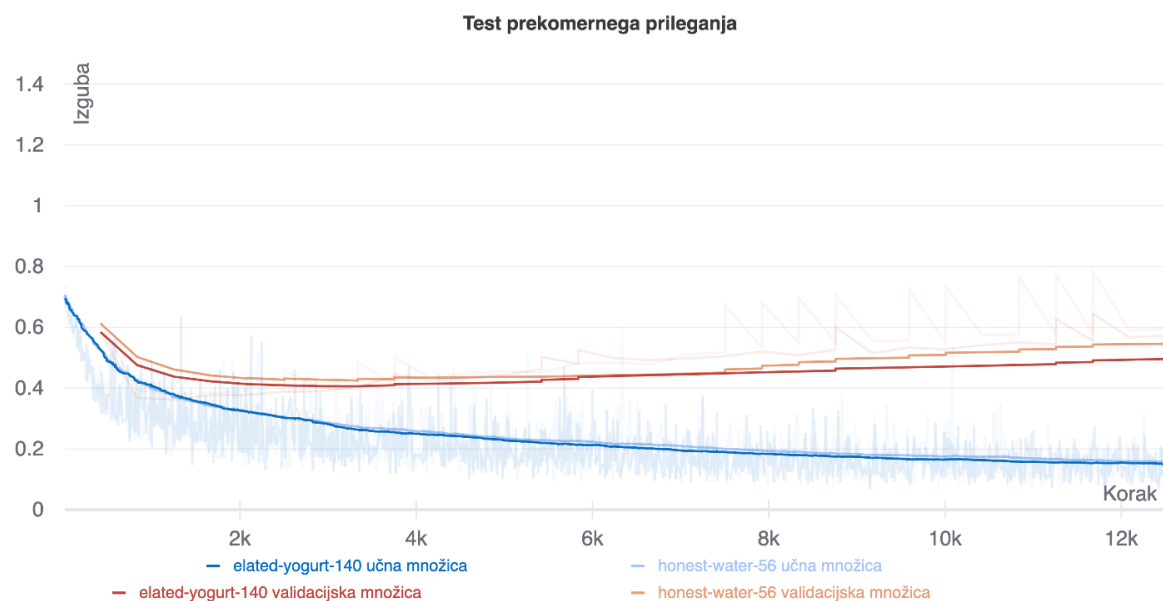
S spremembo podatkov pri modelu EfficientNet-B0 se ne doseže bistvenih sprememb. Kot je nakazano v preglednici 15 je v metriki AUC največja sprememba le za 0,3 % (slika 41). Doseže pa se nekoliko boljšo vrednost izgube, ki se zmanjša za vrednost 0,132 (slika 42). Tako kot v prvem delu testiranj, na določenem koraku model ne pridobiva več dodatnih informacij oziroma novih znanj.

Preglednica 15: Rezultati drugega dela testiranj - EfficientNet-B0

Ime eksperimenta	S2 produkt	Globina	Izguba	AUC [%]
honest-water-56	L1C	8	0,802	80,6
faithful-cloud-136	L1C	15	0,754	79,7
firm-night-138	L2A	10	0,624	80,5
elated-yogurt-140	L2A	16	0,670	80,9



Slika 41: Grafični prikaz metrike AUC po korakih učenja v drugem delu testiranj - EfficientNet-B7



Slika 42: Potek vrednosti izgube pri dveh najboljših testih - EfficientNet-B0

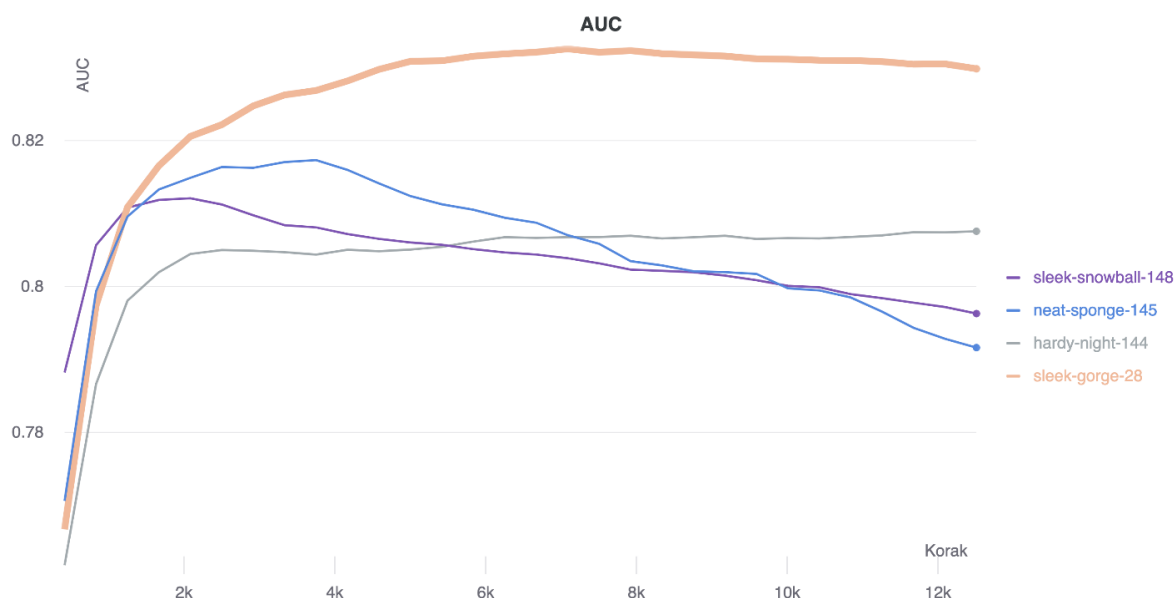
3. EfficientNet-B7

V drugem delu testiranja spremembe podatkovne strukture za model EfficientNet-B7 iz preglednice 16 ugotovim, da se s spremembo podatkovne strukture izgubi na kakovosti modelov. Padec AUC je v vseh primerih približno 3 %. Vrednost izgube se v dveh primerih še poveča, pade le v drugem primeru, vendar je padec AUC prevelik za nadaljnjo obdelavo. Prav tako je iz grafa na Slika 43 razviden potek metrike AUC po korakih, kjer ima predhodni primer dober potek krivulje, medtem ko se ostali primeri obnašajo nenavadno. Iz poteka sklepamo, da se model, zaradi veliko parametrov, nauči zelo hitro, kar pripelje do velikega prekomernega prilaganja (Slika 44).

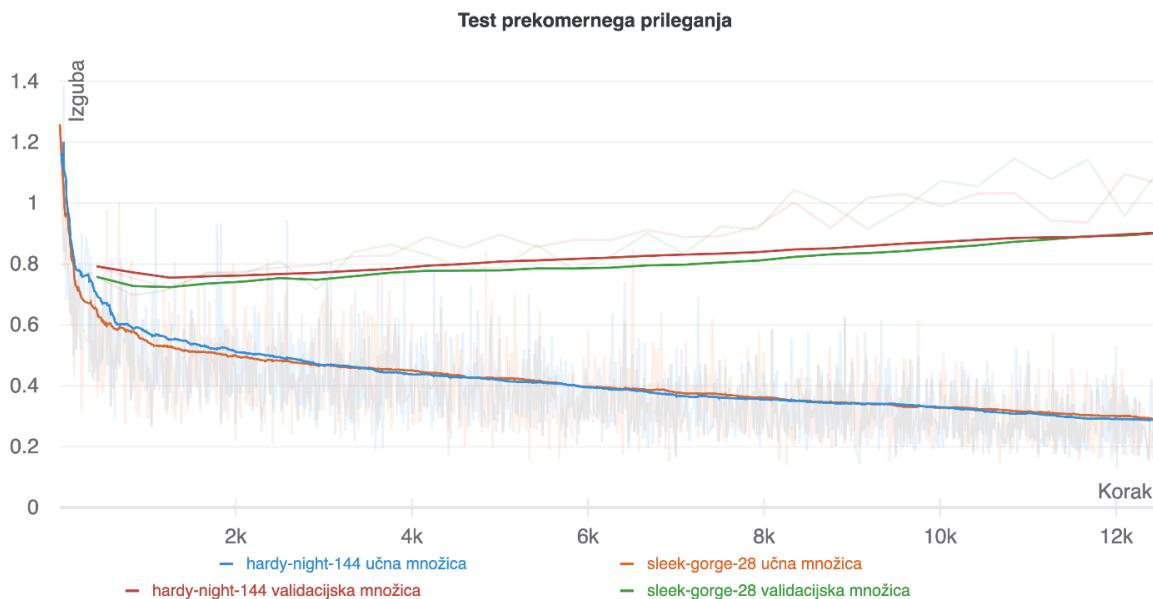
Prekomerno prilaganje, z najvišjo vrednostjo AUC, se pojavi pri obeh testih. Z izboljšanjem podatkov se tega ni odpravilo. Ker je model zelo kompleksen, sem pristopil s poenostavitvijo arhitekture modela in dodajanjem več skritih združevalnih slojev (npr. max pooling).

Preglednica 16: Rezultati drugega dela testiranja - EfficientNet-B7

Ime eksperimenta	S2 produkt	Globina	Izguba	AUC [%]
sleek-gorge-28	L1C	8	1,470	83,0
hardy-night-144	L1C	15	1,354	80,8
neat-sponge-145	L2A	10	1,776	79,2
sleek-snowball-148	L2A	16	1,874	79,6



Slika 43: Grafični prikaz metrike AUC po korakih učenja v drugem delu testiranja - EfficientNet-B7



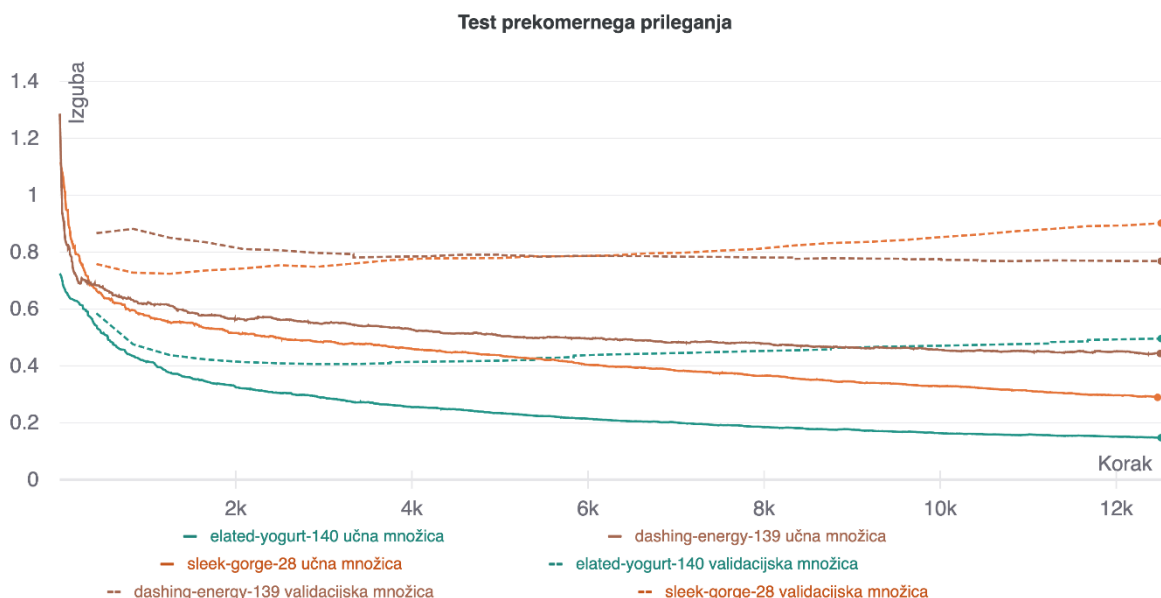
Slika 44: Potek vrednosti izgube pri dveh najboljših testih - EfficientNet-B7

5.2.1 Primerjava med tremi modeli CNN

V preglednici 17 primerjam tri najboljše rezultate modelov, kjer upoštevamo metriko AUC za uspešnost modela in vrednost izgube za prekomerno prileganje. Primerjava pokaže, da prihaja do velikih razlik v vrednosti AUC. Glede na vire raziskav bi se predvidevalo, da se bosta modela EfficientNet odrezala nekoliko bolje, saj sta, v primerjavi s starejšimi modeli, velikokrat dala boljše rezultate. V primeru, ki sem ga analiziral, tega ne morem potrditi, ker prihaja do prevelikih razhajanj v metriki AUC, ki nakazujejo obratni pojav. Prav tako se mora upoštevati, da se modeli različno odzivajo glede na vrsto vhodnih podatkov. MobileNetV2 je pokazal kar 3,6 % boljšo AUC kot EfficientNet-B0 in 1,5 % boljšo AUC od EfficientNet-B7. Vrednosti izgube se lahko primerjajo tudi z močjo modela za napovedovanje (število parametrov), kjer ja zaznati, da se z večanjem števila parametrov modela večja tudi izguba. Pozorni moramo biti tudi na primerjavo vrednosti izgube za učno in validacijsko množico v vsakem modelu (Slika 45). Primerjava kaže, da pri modelu MobileNetV2 do prekomernega prileganja skoraj ne prihaja, medtem ko pri ostalih dveh modelih graf vidno nakaže večanje razlike izgube med učno in validacijsko množico.

Preglednica 17: Metrike treh najuspešnejših modelov drugega dela testiranja

Ime eksperimenta	Model	Izguba	AUC [%]
dashing-energy-139	MobileNetV2	0,729	84,5
elated-yogurt-140	EfficientNet-B0	0,670	80,9
sleek-gorge-28	EfficientNet-B7	1,470	83,0



Slika 45: Test prekomernega prileganja med tremi modeli

5.3 Rezultat testiranja večanja podatkov in števila epoh

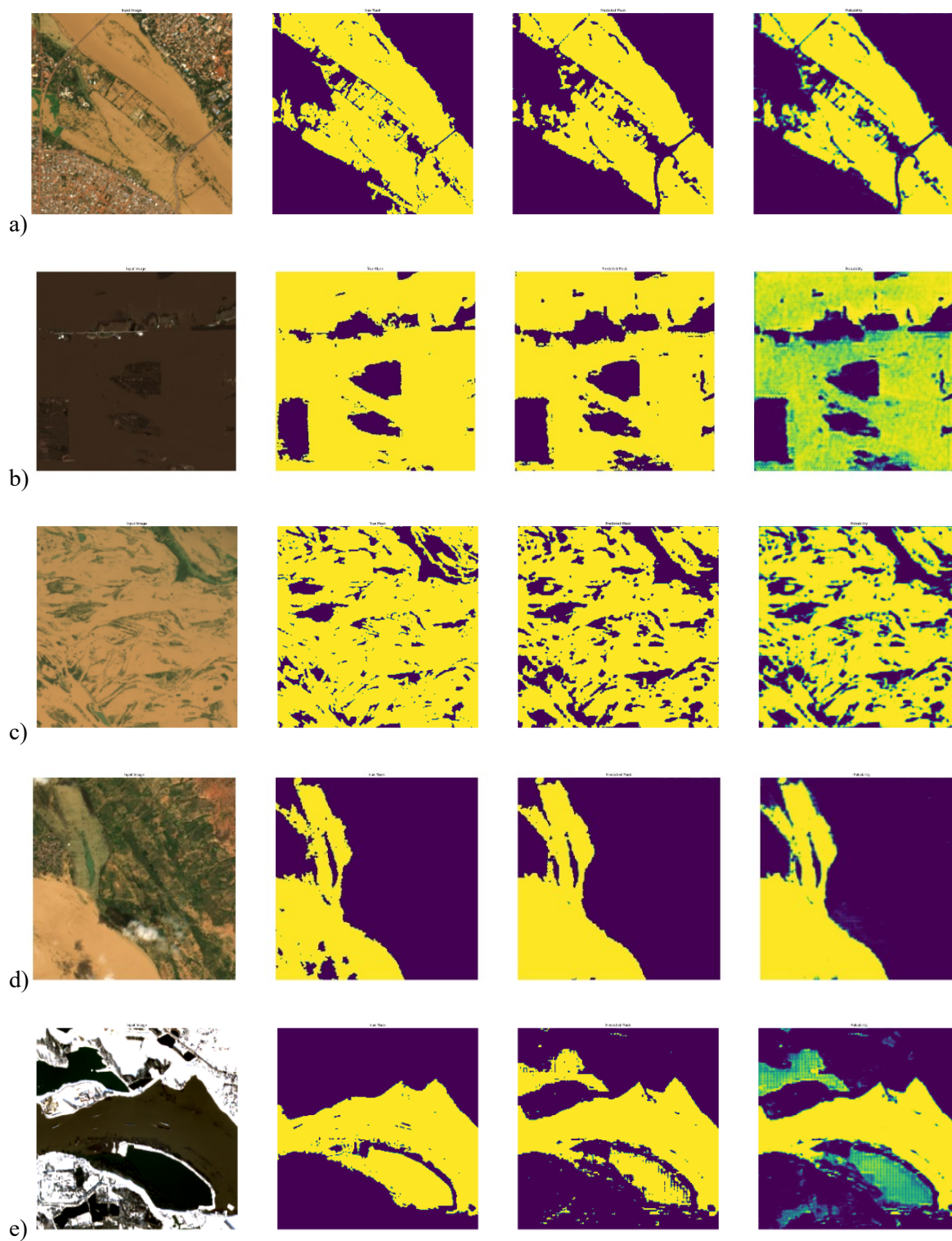
Dodatno sem želel raziskati, ali s povečanjem prekrivanja med posnetki izboljša učenje modela. Začetno prekrivanje je bilo 400 m, naredil sem povečavo na 600 m. Ravno tako sem povečal število epoh iz 30 na 100. V preglednici 18 je prikazano večanje števila korakov učenja in čas učenja. Ob povečanju prekrivanja med posnetki sem dosegel izboljšanje modela za kar 2,3 %. Ob tem se vrednost izgube ni bistveno povečala, le za 0,108. Povečanje števila epoh izraža zadnji prikazan primer v preglednici 18. Ob veliki povečavi števila korakov se posledično poveča tudi čas učenja, vendar se vrednost AUC ne povečuje, ampak začne padati, kar je pojav prekomernega prileganja.

Preglednica 18: Rezultati testiranja večanja podatkov in števila epoh

Ime eksperimenta	Sprememba	Št. korakov učenja	Čas učenja [min]	Izguba	AUC [%]
dashing-energy-139	brez spremembe	12510	80	0,729	84,5
wandering-sea-151	povečano prekrivanje	21030	140	0,837	86,8
misunderstood-paper-153	povečano prekrivanje in število epoh	70100	440	1,188	84,9

Na sliki 46 prikažem nekaj primerov, ki so naključno izbrani iz validacijske množice naučenega modela wandering-sea-151. Vsak primer je ponazorjen na štirih slikah, kjer prva prikazuje izvorno sliko v prikazu RGB, druga je maska pravih vrednosti, tretja je maska napovedanih vrednosti in zadnja je maska

verjetnosti. Prvi trije primeri (slika 46 a, b, c) prikazujejo uspešno napovedane vrednosti. Primer d) uspešno napove vrednosti, vendar ima težavo prepoznavne sence oblakov. Podobna težava se pojavi tudi na nekaterih drugih posnetkih, kar privede do sklepa, da je naučen model občutljiv na sence oblakov. Zadnji primer e) napove poplavna območja bolje kot je določena binarna maska pravih vrednosti. Razvidno je, da na zgornjem in spodnjem delu pravih vrednosti, manjkata označeni vodni območji. To je najboljše očitno na sliki verjetnosti, kjer se opazijo vsa vodna območja.



Slika 46: Satelitski posnetek (RGB), maska pravih podatkov, maska napovedanih vrednosti, verjetnost napovedanih vrednosti (od levega do desnega posnetka)

6 UPORABA MODELA NA NOVIH PODATKIH

V eksperimentalnem delu sem izdelal model z natančnostjo 86,8 % po metriki AUC, kar velja za uspešen model učenja. Neglede na analizo učenja na validacijskih podatkih, sem želel model predstaviti na novem primeru, neodvisnem od učnih podatkov. Območje sem izbral glede na poročilo poplavljanja rek, med 15. in 21. septembrom 2017, Agencije RS za okolje (Agencija RS za okolje, 2017). Pregledal sem določene poplave in razpoložljive satelitske posnetke Sentinel-2 po datumih ter izbral poplavo na reki Krki na območju, kjer se iz satelitskih posnetkov poplava najboljše prepozna. Okviren centroid območja ima koordinate (geografska dolžina, geografska širina) = (15,361102°; 45,873432°).

Naučeni model sem naložil v program GMS-GIS in določil podatkovne parametre za učenje na novih podatkih. Z vgrajenim orodjem za napovedovanje sem opravil celoten postopek. Najprej sem z izbranim območjem interesa narisal poligon, določil začetni in končni datum ter naučeni model. Na začetku procesiranja program poišče vse razpoložljive satelitske posnetke in v časovnem parametru, ki sem ga opredelil z datumi. Na vseh posnetkih opravi obdelavo prepoznavne poplav z naučenim modelom.

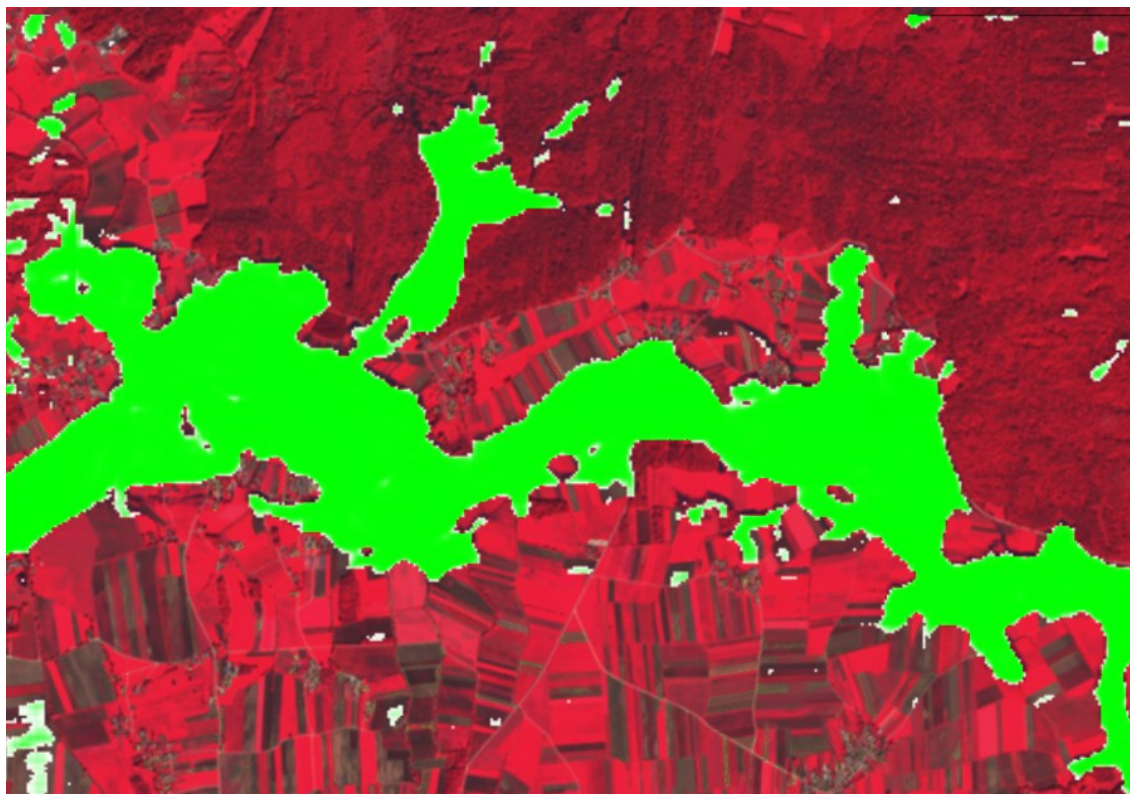
Na naslednjih slikah je prikazana razlika med ne poplavljenim in poplavljenim območjem ter rezultat napovedi. Slika 47 prikazuje stanje na območju z rečno strugo na dan 8. 9. 2017, kjer je razvidno, da je območje ne poplavljen. Slika 48 prikazuje območje s poplavo na dan 18. 9. 2017. Napoved poplave prikazuje slika 49, kjer je z barvno lestvico, od bele proti rdeči, prikazana verjetnost poplave (zelena = večja verjetnost).



Slika 47: Satelitski posnetek v času nepopravljenega območja in prikaz struge v tem času



Slika 48: Satelitski posnetek v času poplavljenega območja in prikaz struge v tem času



Slika 49: Satelitski posnetek v času poplavljenega območja in prikaz napovedane vrednosti poplavljenega območja

7 ZAKLJUČEK

Magistrsko delo zajema korake od izdelave lastne zbirke podatkov za strojno učenje do uporabe le te in prikaza rezultatov opravljenih testov. V veliki meri se osredotočim na modele strojnega učenja in prilagajanje le-teh, za dosegajo najboljših rezultatov pri napovedovanju poplavnih območij. Zastavil sem si načrt analiz in potek testiranja po delih. Idejno izhodišče naloge je ustvariti program za optimizacijo modelov strojnega učenja v tolikšni meri, da je uporaben za uspešno prepoznavanje poplavnih območij iz satelitskih posnetkov Sentinel-2. Ustvaril sem zbirko poplavnih območij, s pripravo surovih podatkov satelitskih posnetkov in mask, ki ločujejo poplavna od ostalih območij. Maske območja, na katerih je območje označeno kot poplavno meri kar 7884 km², to je osnovna dimenzija zajetega območja. V času testiranja sem z različnimi tehnikami povečevanja podatkov dimenzijo znatno povečal.

Ob zaključevanju programa za generiranje podatkov, sem po poplavnih dogodkih ločeno shranil surove podatke in maske. Generiral sem štiri različne kombinacije spektralnih in analitičnih kanalov, ki so opredeljeni v poglavju 3.2. Po izdelavi podatkov sem uporabil program, v katerega smo vgradili tudi modele strojnega učenja. Sledila je izvedba testiranja, kjer sem v prvem delu testiranja analiziral hiperparametre po modelih. Te sem obravnaval za vsak model posamično, ker se optimalna kombinacija hiperparametrov razlikuje glede na model. Iz analize prvega dela testiranja sem ugotovil, da se za model MobileNetV2 uporabi rmsprop optimizator, dice kriterijsko funkcijo in 0.001 za stopnjo učenja, kar zagotovi rezultat 83,1 % AUC. Pri modelu EfficientNet-B0 sem ugotovil, da so najustreznejši optimizator adam, bce kriterijska funkcija in 0.0001 stopnja učenja, kjer sem dobil rezultat 80,6 % AUC. Za zadnji model EfficientNet-B7 so primerni rmsprop optimizator, kriterijska funkcija dice in 0.0001 stopnja učenja pri 83 % AUC. S prvim delom testiranja sem potrdil, da se primerni hiperparametri za posamezen model razlikujejo, zato sem se v prvem delu odločil za tri rezultate – od vsakega modela po enega. Pri razlikah v AUC v modelu sem ugotovil, da je največja razlika pri MobileNetV2, in sicer 10 %, medtem ko je pri EfficientNet-B0 in EfficientNet-B7 4 % in 2,5 %. Iz tega sem sklepal, da sta druga dva modela manj občutljiva na spreminjanje hiperparametrov. Pri primerljivih rezultatih je EfficientNet-B0 nekoliko slabši.

Drugi del testiranja se nanaša na spreminjanje podatkovne strukture, kako vplivajo na učenje modelov, in analizo. Za osnovo sem uporabil rezultate prvega dela testiranja, ki je sestavljena iz šest izbranih spektralnih kanalov, maske oblakov in maske brez podatkov. Nato sem podatke izboljšal z večanjem spektralnih kanalov in uporabo Sentinel-2 L2A podatkov, ki imajo odpravljen vpliv svetlobnega odboja atmosfere. Izboljšati sem želel ne le najuspešnejši model prvega dela testiranja, ampak tudi izboljšati rezultat vsakega modela, saj spreminjanje podatkov različno vpliva na različne modele. Znatno izboljšanje sem dobil pri modelu MobileNetV2, kjer se vrednost AUC izboljša za kar 1,4 %. Za 0,3 % sem pridobil na modelu EfficientNet-B0, medtem ko imamo pri modelu EfficientNet-B7 že najboljši rezultat oziroma s spreminjanjem znižamo vrednost AUC. V kolikor se izvzame zadnji model, je v prvih dveh pričakovana izboljšava pri zadnji podatkovni kombinaciji. Vsebuje vse spektralne kanale Sentinel-

2, popravljene za vpliv atmosfere, natančnejšo masko oblakov, masko snežnih površin v zasneženem pobočju. Na koncu drugega dela testiranja sem naredil primerjavo med tremi najuspešnejšimi modeli testiranja spremembe podatkov. Izbral sem model MobileNetV2, ki ima najvišjo vrednostjo AUC. Prav tako je bil potreben pregled prekomernega prileganja učni množici, ki je pri izbranem modelu najmanjši, kot je razvidno iz dodanega grafa. Razlika med vrednostjo izgube učne in validacijske množice se pri izbranem modelu, za razliko od drugih dveh modelov, ne povečuje v odvisnosti od koraka učenja.

V zadnjem delu testiranja sem opravil dodatna testa. Kot osnovo sem uporabil rezultat drugega dela testiranja. Prva sprememba je bila povečanje podatkov z večanjem prekrivanja med posnetki ob generiranju posnetka, druga pa povečanje števila epoh učenja. Najboljši rezultat sem dosegel z večanjem količine podatkov, kjer sem vrednost AUC povečal za 2,3 % ter s tem dosegel najvišjo uspešnost modela MobileNetV2, ki ima vrednost AUC 86,8 %.

V nalogi sem potrdil oziroma delno potrdil zastavljene hipoteze.

Prvo hipotezo potrdim, ker z izdelanim modelom konvolucijskih nevronske mreže lahko v zadostni meri prepoznam popravna območja.

V drugi hipotezi zastavljeno trditev, da je izbira podatkovnega produkta satelita Sentinel-2 L2A primernejša od L1C, potrdim, saj v rezultatih drugega dela testiranja dobim potrditev.

V tretjem delu testiranja sem delno zavrnil tretjo hipotezo, da se lahko z večanjem števila epoh doseže večja vrednost AUC modela. V eksperimentalnem delu je dokazano, da se kakovost napovedovanja modela zmanjšuje, ker prihaja do prekomernega prileganja, kar pomeni, da se je optimalno število epoh že doseglo. Drugi del tretje hipoteze, da izbrni model s hiperparametri izboljšam po metriki AUC s tehnikami bogatenja količine podatkov, potrdim.

Slika 46 prikazuje rezultate na validacijski množici, iz katere je razvidno, da so napovedi skoraj na vseh dodanih primerih uspešne. Na primeru e) slika prikazuje, da je model bolje prepoznal območja, kot je bila narejena maska. Iz primera d) je zaznati problem, in sicer sence oblakov. Iz tega lahko povzamemo naslednje izboljšave na podatkovni ravni. Ponoven pregled maskiranega območja poplav in izboljšava le-tega, prav tako večanje območja, evidentira več poplavnih dogodkov. Glede na kompleksnost modelov sem operiral z relativno malo količino podatkov.

Ugotovil sem, da je naučeni model občutljiv na sence, da bi se lahko z vgradnjo maske senc oblakov še optimiziral. Produkt Sentinel-2 L2A ponuja tudi sloj območne klasifikacije, iz katere bi lahko pridobil masko senc oblakov. Dodatno se lahko izboljša tudi arhitekturo modela. V uporabljene pre-definirane modele bi se vgradili združevalni sloji in tako poenostavili model. To je izvedljivo na podlagi moje ugotovitve, da se pojav prekomernega prileganja učnim podatkom pojavi pri kompleksnejših modelih, predvsem pri modelu EfficientNet-B7.

Primer uporabe modela v GIS-u le potrjuje uspešnost naučenega modela in njegovo uporabnost.

8 VIRI

Agencija RS za okolje. (2017). *Poplavljanje rek med 15. in 21. septembrom 2017*.

Amrutha, J., & Remya Ajai, A. S. (2018). Performance analysis of backpropagation algorithm of artificial neural networks in verilog. *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, RTEICT 2018 - Proceedings*. <https://doi.org/10.1109/RTEICT42901.2018.9012614>

Bazi, Y., al Rahhal, M. M., Alhichri, H., & Alajlan, N. (2019). Simple Yet Effective Fine-Tuning of Deep CNNs Using an Auxiliary Classification Loss for Remote Sensing Scene Classification. *Remote Sensing*, 11(24). <https://doi.org/10.3390/rs11242908>

Dong, K., Zhou, C., Ruan, Y., & Li, Y. (2020, December). MobileNetV2 Model for Image Classification. *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*. <https://doi.org/10.1109/ITCA52113.2020.00106>

ESA. (2018). *Sen2Cor Configuration and User Manual*. https://step.esa.int/thirdparties/sen2cor/2.5.5/docs/S2-PDGS-MPC-L2A-SUM-V2.5.5_V2.pdf

ESA. (2021a). *ESA Product types*. <https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/product-types>

ESA. (2021b). *Sentinel-2*. Sentinel Online. <https://sentinel.esa.int/web/sentinel/missions/sentinel-2>

ESA. (2021c). *Sentinel-2 Level-1*. Sentinel Online. <https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/product-types/level-1c>

ESA. (2021d). *Sentinel-2 Level-2*. <https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/product-types/level-2a>

European Space Agency. (2021). *SNAP*. <http://step.esa.int/main/toolboxes/snap/>

European Union. (2021). *The Copernicus Emergency management Service - Mapping*. <https://emergency.copernicus.eu/mapping/search/node/flood>

Flycom Technologies d.o.o. (2021). *GMS* (5.9.8).

Goffi, A., Stroppiana, D., Brivio, P. A., Bordogna, G., & Boschetti, M. (2020). Towards an automated approach to map flooded areas from Sentinel-2 MSI data and soft integration of water spectral features. *International Journal of Applied Earth Observation and Geoinformation*, 84. <https://doi.org/10.1016/j.jag.2019.101951>

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. Volume 1. In *Nature* (Vol. 29, Issue 7553).

HERE. (2021). *HERE*. <https://developer.here.com/products/platform/data>

- Jadon, S. (2020). *A survey of loss functions for semantic segmentation*.
<https://doi.org/10.1109/CIBCB48159.2020.9277638>
- JetBrains d.o.o. (2021a). *DataGrip* (2021.2.4).
- JetBrains d.o.o. (2021b). *PyCharm* (2021.2.4).
- Karimi, A. (2021). *Instance vs Batch Normalization*. <https://www.baeldung.com/cs/instance-vs-batch-normalization>
- Keijsers, N. L. W. (2010). Neural Networks. *Encyclopedia of Movement Disorders*, 257–259.
<https://doi.org/10.1016/B978-0-12-374105-9.00493-7>
- Keras. (n.d.). *Keras Applications*. Retrieved September 10, 2021, from <https://keras.io/api/applications/>
- Kingma, D. P., & Ba, J. (2014). *Adam: A Method for Stochastic Optimization*.
<https://arxiv.org/pdf/1412.6980.pdf>
- Kononenko, I., & Kuhar, M. (2007). Measures for performance evaluation. In *Machine Learning and Data Mining* (pp. 74–78). Horwood Publishing.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90. <https://doi.org/10.1145/3065386>
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11). <https://doi.org/10.1109/5.726791>
- Li, W., Fu, H., Yu, L., Gong, P., Feng, D., Li, C., & Clinton, N. (2016). Stacked Autoencoder-based deep learning for remote-sensing image classification: a case study of African land-cover mapping. *International Journal of Remote Sensing*, 37(23), 5632–5646.
<https://doi.org/10.1080/01431161.2016.1246775>
- Li, X., Sun, X., Meng, Y., Liang, J., Wu, F., & Li, J. (2019). *Dice Loss for Data-imbalanced NLP Tasks*.
- Li, Z., & Arora, S. (2019). *An Exponential Learning Rate Schedule for Deep Learning*.
- Lin, M., Chen, Q., & Yan, S. (2013). *Network In Network*. <http://arxiv.org/abs/1312.4400>
- Main-Knorn, M., Pflug, B., Louis, J., Debaecker, V., Müller-Wilm, U., & Gascon, F. (2017). Sen2Cor for Sentinel-2. In L. Bruzzone, F. Bovolo, & J. A. Benediktsson (Eds.), *Image and Signal Processing for Remote Sensing XXIII*. SPIE. <https://doi.org/10.1117/12.2278218>
- Michael A. Nielsen. (2015). *Neural Networks and Deep Learning*.
<http://neuralnetworksanddeeplearning.com/>
- Michele, A., Colin, V., & Santika, D. D. (2019). MobileNet Convolutional Neural Networks and Support Vector Machines for Palmprint Recognition. *Procedia Computer Science*, 157.
<https://doi.org/10.1016/j.procs.2019.08.147>

- OpenLayers. (2021). *Region Growing*. OpenLayers. <https://openlayers.org/en/latest/examples/region-growing.html>
- Pratt, W. K. (2007). *Digital Image Processing 4th Edition* (4th ed.).
- Rad, A. M., Kreitler, J., & Sadegh, M. (2021). Augmented Normalized Difference Water Index for improved surface water monitoring. *Environmental Modelling & Software*, 140. <https://doi.org/10.1016/j.envsoft.2021.105030>
- Ramachandran, P., Zoph, B., & Le, Q. v. (2017). *Searching for Activation Functions*.
- Ramachandran, P., Zoph, B., & Le, Q. v. (2018). SWISH: A SELF-GATED ACTIVATION FUNCTION. *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings, 1*.
- Repse, M., Lubej, M., & Zupanič, A. (2020). Cloud Masks at Your Service. *Medium*. <https://medium.com/sentinel-hub/cloud-masks-at-your-service-6e5b2cb2ce8a>
- Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. <https://arxiv.org/pdf/1505.04597.pdf>
- Rosebrock, A. (2022). *Intersection over Union (IoU) for object detection*. <https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. <https://doi.org/doi.org/10.48550/arXiv.1801.04381>
- Sinergise d.o.o. (2020). *EO Browser*. SentinelHub. <https://www.sentinel-hub.com/explore/eobrowser/>
- Sinergise d.o.o. (2021a). *Sentinel Hub*.
- Sinergise d.o.o. (2021b). *SentinelHub*. SentinelHub. <https://www.sentinel-hub.com/about/>
- Tamir, M. (2020). What Is Machine Learning? *UC Berkeley*. <https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/>
- Tan, M., & Le, Q. v. (2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. <https://arxiv.org/pdf/1905.11946.pdf>
- Tappert, C. C. (2019, December). Who Is the Father of Deep Learning? *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*. <https://doi.org/10.1109/CSCI49370.2019.00067>
- The International Charter Space and Major Disasters. (n.d.). *The International Charter Space and Major Disasters*. Retrieved February 22, 2022, from <https://disasterscharter.org/web/guest/home>

- Tian, Y. (2020). Artificial Intelligence Image Recognition Method Based on Convolutional Neural Network Algorithm. *IEEE Access*, 8, 125731–125744. <https://doi.org/10.1109/ACCESS.2020.3006097>
- Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the Gradient by a Running Average of Its Recent Magnitude. In *COURSERA: Neural Networks for Machine Learning* (pp. 26–31). <https://docs.google.com/viewer?url=http%3A%2F%2Fwww.cs.toronto.edu%2F~hinton%2Fcoursa%2Flecture6%2Flec6.pdf>
- Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2016). *Instance Normalization: The Missing Ingredient for Fast Stylization*.
- Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2017). Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-January*. <https://doi.org/10.1109/CVPR.2017.437>
- Varone, M., Mayer, D., & Melegari, A. (2020). *What is Machine Learning? A definition*. Expert System. <https://www.expert.ai/blog/machine-learning-definition/>
- Wikipedia. (2021). *List of deadliest floods*. https://en.wikipedia.org/wiki/List_of_deadliest_floods
- Xu, H. (2006). Modification of normalised difference water index (NDWI) to enhance open water features in remotely sensed imagery. *International Journal of Remote Sensing*, 27(14). <https://doi.org/10.1080/01431160600589179>
- Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. In *Insights into Imaging* (Vol. 9, Issue 4, pp. 611–629). Springer Verlag. <https://doi.org/10.1007/s13244-018-0639-9>