Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Tim Smole

Globoko učenje za segmentacijo in klasifikacijo cestišča

magistrsko delo

magistrski program druge stopnje
računalništvo in informatika

Mentor: izr. prof. dr. Danijel Skočaj

Ljubljana, 2019

Zahvala

Zahvaljujem se mentorju izr. prof. dr. Danijelu Skočaju za pomoč in vodstvo pri izdelavi naloge, podjetju DFG Consulting za dostop do podatkovne zbirke, sestri za lektoriranje, staršem za spodbudo in vsem ostalim, ki so na kakršenkoli način pomagali pri izdelavi magistrske naloge.

Tim Smole, 2019

Vsem.

"Any AI smart enough to pass a Turing test is smart enough to know to fail it."

— Ian McDonald, River of Gods

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
CNN	convolutional neural network	konvolucijska nevronska mreža
FCN	fully convolutional network	popolnoma konvolucijska mreža
MSI	modified Swiss index	modificirani Švicarski indeks
CSF	Cityscape dataset with fina annotations	podatkovna zbirka Cityscape z natančno označenimi podatki
CSC	Cityscape dataset with coarse annotations	podatkovna zbirka Cityscape z grobo označenimi podatki
CSCE	Cityscape dataset with coarse extra annotations	podatkovna zbirka Cityscape z dodatnimi grobo označenimi podatki
RSTS	segmentation dataset for road surface type	podatkovna zbirka za segmentacijo tipa cestišča
RST	classification dataset for road surface type	podatkovna zbirka za klasifikacijo tipa cestišča
FP	false positive	lažno pozitivni primer
TP	true positive	resnično pozitivni primer
FN	false negative	lažno negativni primer
TN	true negative	resnično negativni primer
FPR	false positive rate	lažna pozitivna stopnja
CA	classification accuracy	klasifikacijska točnost
SSD	sum of squared difference	vsota kvadratnih razlik

Povzetek

Naslov: Globoko učenje za segmentacijo in klasifikacijo cestišča

Eden izmed problemov, s katerim se srečujejo vzdrževalci cestišč, je zahteva po rednem posodabljanju evidence o kvaliteti vozišč. Podatke o poškodbah trenutno beležijo ročno, kar pa je časovno zamudno in pogosto nekonsistentno.

V nalogi smo s tem razlogom predstavili pristop k podobnemu problemu, kjer namesto poškodovanosti vozišča avtomatično določamo tip površine. Za reševanje tega problema smo uporabili klasifikacijsko umetno nevronsko mrežo, ki temelji na arhitekturi ResNet-50. Da pa bi izboljšali njeno uspešnost, smo v vhodne slike vkomponirali informacijo o položaju cestišča, pridobljeno s segmentacijsko mrežo U-Net. Pokazali smo, kako lahko v primeru segmentacije uporabimo informacijo o položaju robov cestišča in slikovnim elementom v neposredni bližini dodelimo večjo utež ter s tem usmerimo pozornost mreže v dele slike, kjer se napake najpogosteje nahajajo. Pokazali smo tudi, kako v primeru delno označenih podatkov uporabimo neoznačene dele slike, jim dodelimo nižjo utež in jih nato upoštevamo v času učenja.

Primerjali smo tudi dva pristopa k usmerjanju pozornosti klasifikacijskih mrež. Prvi pristop uporablja maskiranje vhodne slike z ničelno vrednostjo, kjer je segmentacijska mreža detektirala ozadje, drugi pa temelji na razširitvi vhodne slike z izhodom segmentacijske mreže. Pokazali smo, da se uporaba informacije o položaju cestišča s pomočjo segmentacije obrestuje, saj se mera uspešnosti F1 pridobljena na testni zbirki poveča iz 0,947 na 0,971, v kolikor uporabimo slednji pristop.

Ključne besede

nevronske mreže, segmentacija, klasifikacija, usmerjanje pozornosti, cestišča, računalniški vid

Abstract

Title: Deep learning for road segmentation and classification

One of the problems road holders are facing is maintaining a record of road's surface quality. They acquire a vast amount of image data and then assess the surface quality by manually inspecting those images, which is time consuming and often inconsistent.

In this work we show how to tackle a similar problem of automatic recognition of road surface type. To solve this problem we use the artificial neural network for classification tasks based on ResNet-50 architecture. To boost it's performance we use the information of the road's position in the input image which is obtained with U-Net neural network for semantic segmentation. In case of segmentation we show how to emphasize pixels located near road's edges and focus the network's attention during training to the parts where errors are most frequent. We also consider coarsely annotated images and show how we can use unlabeled pixels assigning them lower weights during the training process.

We compare two attention mechanisms for neural networks used for classification tasks. The first mechanism masks input images with zero values where segmentation network detects background. The second mechanism is based on extending the input image with an output of U-Net. We show that by using the second approach F1 score evaluated on the test dataset improves from 0.947 to 0.971.

Keywords

neural networks, segmentation, classification, attention mechanism, roads, computer vision

Poglavje 1

Uvod

1.1 Motivacija

V zadnjem stoletju je v zahodni družbi avto postal del vsakdana. Število vozil se iz dneva v dan povečuje, hkrati pa smo na prelomni točki zgodovine, ko bodo avtonomna vozila postala nekaj običajnega. Za uspešno avtonomnost in varno navigacijo v prostoru, kateri smo danes priča, so v glavnem zaslužni čedalje zanesljivejši senzorji, še bolj pa nedavni napredki na področju obdelave enormne količine podatkov, ki jih senzorji proizvajajo. Če za trenutek odmislimo vse prometne predpise in nepredvidljive faktorje, kot so pešci in vozila, lahko opazimo, da je za varno avtonomno vožnjo ključnega pomena kvalitetna in dobro vzdrževana cesta. Uporabniki cestišč to marsikdaj odmislimo, za samovozeča vozila pa je zanesljiva detekcija cestišča in njenega stanja bistvenega pomena.

Avtonomna vozila pa niso edina, ki zahtevajo temeljito analizo cestišč. Potreba po podobnem prepoznavanju cestnih poškodb in celovitem pregledu stanja cestišč prihaja tudi s strani njihovih lastnikov in vzdrževalcev. Tako kot število vozil se iz dneva v dan povečuje tudi število cestišč, kar za vzdrževalce posledično predstavlja težji nadzor nad obrabo in poškodovanostjo posameznih odsekov vozišč.

V praksi se poškodbe cestišč običajno ocenjujejo po metodi modificiranega

Slika 1.1: Primer vozila za periodično zajemanje slik cestišča [1].

švicarskega indeksa (v nadaljevanju MSI), ki upošteva štiri vrste poškodbe: razpoke, obrabo, udarne jame in krpe. MSI se izračuna tako, da za vsako od štirih kategorij vizualno oceni jakost in obseg poškodbe za 50 metrski odsek cestišča.

V preteklosti so vzdrževalci popis poškodb opravljali ročno, pri čemer so na terenu opisali ali skicirali poškodbe in jih kasneje vnesli v podatkovno bazo, ki je služila nadaljnjim analizam. Proces zajemanja informacij o poškodbah je bil v preteklih letih do velike mere avtomatiziran. Danes imajo vzdrževalci vozila, opremljena s specializiranimi kamerami, ki periodično zajemajo slike cestišč. Primer takega vozila je prikazan na Sliki 1.1. Z njimi so si vzdrževalci pridobili ogromno podatkovno zbirko slik cestnih odsekov, ki jih sedaj ročno pregledujejo in ocenjujejo njihove poškodbe.

Tako ocenjevanje ni le naporno in zamudno za vzdrževalce, temveč zna biti tudi neobjektivno, saj dva različna ocenjevalca zaradi lastne presoje lahko isto poškodbo ocenita na drugačen način. Zamudnost in neobjektivnost ocenjevanja pa sta samo dva od mnogih razlogov, ki kličeta po avtomatizaciji ocenjevalnega postopka.

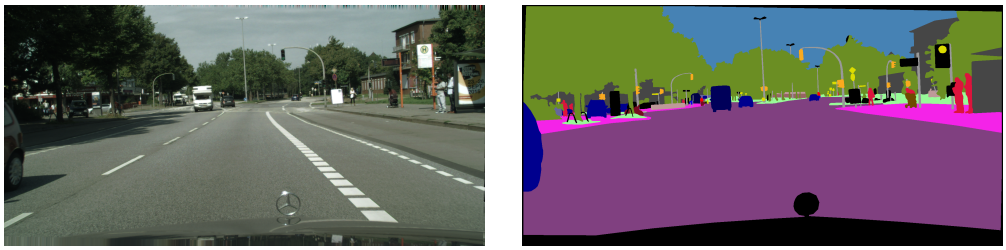
1.2 Umestitev problema

Zasluge, da se lahko učinkovito soočimo s problemom avtomatičnega ocenjevanja poškodb vozne površine, gredo nedavnim prebojem na področju računalniškega razumevanja slikovne vsebine. Na tem področju je zaslovela družina algoritmov t.i. umetnih nevronskih mrež oziroma posebna skupina nevronskih mrež, katerih prevladujoča operacija je konvolucija in jih zato pogosto imenujemo konvolucijske nevronske mreže (angl. Convolutional Neural Network, ali krajše CNN).

Problematiko računalniškega razumevanja slikovne vsebine lahko v grobem razdelimo v tri kategorije: klasifikacija, detekcija oziroma lokalizacija in segmentacija. Pri klasifikaciji slik običajno želimo sliko uvrstiti v eno izmed vnaprej znanih kategorij. Pri detekciji oziroma lokalizaciji želimo ugotoviti, ali se in kje na sliki se določen objekt nahaja. Segmentacija od vseh treh kategorij predstavlja najpodrobnejše razumevanje slikovne vsebine. Tu napovedujemo, kateri od vnaprej znanih kategoriji pripada posamezni slikovni element. Lahko bi rekli, da gre za klasifikacijo na ravni slikovnih elementov.

Potrebno je razumeti, da ključno vlogo za uspešno učenje in delovanje umetnih nevronskih mrež igrajo učni podatki. Brez le-teh nam konvolucijske nevronske mreže ne bi kaj dosti pomagale. Učni podatki so sestavljeni iz množice učnih primerov, kjer slika in sliki pripadajoča oznaka predstavljata en učni primer. Kakšna je oznaka pa je odvisno od problema, ki ga želimo rešiti. V primeru klasifikacije oznako predstavlja kategorijo, v katero spada pripadajoča slika. Ko govorimo o lokalizaciji, je oznaka običajno predstavljena s štirimi parametri - x in y koordinati ter višina in širina najdenega objekta, v primeru detekcije pa še kategorija, v katero spada najden objekt. Segmentacijska oznaka pa je pravzaprav maska istih dimenzij, kot jih ima njej pripadajoča slika, kjer pa vrednosti elementov predstavljajo kategorije istoležnih slikovnih elementov pripadajoče slike. Primer segmentacijske maske je prikazan na Sliki 1.2.

Problem prepoznave kvalitete cestišča lahko uvrstimo med klasifikacijske probleme, saj posamezni sliki želimo določiti vrednost, ki predstavlja



Slika 1.2: Primer učnega primera za semantično segmentacijo vzet iz zbirke Cityscape [2]. Levo vhodna slika in desno segmentacijska maska, kjer vsaka barva ponazarja eno od kategorij.

kvaliteto cestišča prikazanega na tej sliki. A ker običajno velik del slike ne predstavlja vozišča, bi zajeten del vhoda v nevronsko mrežo predstavljal šum. Zato lahko celotni problem razdelimo na dva podproblema. Naša prva naloga je zaznati, kaj predstavlja cesto in kaj ozadje, pri čemer je ozadje definirano kot vse, kar ni cesta. Rešitev prvega problema nam nato predstavlja informacijo o položaju cestišča, ki jo lahko uporabimo za usmerjanje pozornosti klasifikacijske mreže, s katero napovedujemo kvaliteto vozne površine.

Prvi podproblem bi lahko obravnavali kot lokalizacijski problem, saj moramo določiti, kje na sliki se nahaja cesta. V tem primeru bi bil izhod iz nevronske mreže omejitveni okvir (angl. bounding box), v katerem se nahaja cesta. A ker lahko pričakujemo, da ceste niso vedno pravokotne oblike in se na cestah pogosto nahajajo tudi kakšne ovire, kot na primer vozila, je problem bolj primerno obravnavati kot segmentacijski problem. Na ta način se resnično znebimo odvečne informacije na sliki, ostane pa nam le del slike, ki predstavlja cesto. Za učenje nevronske mreže v tem primeru poleg slik potrebujemo še njihove segmentacijske maske. Ker se pločniki ne razlikujejo bistveno od cestišča, imajo nevronske mreže v praksi pogosto težavo z razlikovanjem robnih delov cestišča. V kolikor bi radi usmerili pozornost nevronske mreže na robove vozišča, je preciznost segmentacijskih mask ključnega pomena.

Na žalost pa vzdrževalci cestišč takih mask običajno nimajo, saj je njihova priprava zelo zahtevna in časovno potratna. Raziskava [3] je pokazala,

da ročno označevanje enega samega objekta na nivoju posameznih slikovnih elementov v povprečju traja približno 80 sekund, medtem ko označevanje na nivoju slike (torej označevanje, primerno za klasifikacijo), traja le kakšno sekundo. Avtorji članka [2] pa poročajo, da označevanje celotne slike na nivoju slikovnih elementov zahteva uro in pol človeškega truda - vključno s kontrolo kvalitete. Za uspešno učenje nevronskih mrež običajno potrebujemo tudi po več tisoč učnih slik, zato se poraja vprašanje, ali obstaja kakšna bližnjica, ki bi izpustila ali pa vsaj skrajšala potratno označevanje posameznih slikovnih elementov. Čas priprave segmentacijskih oznake lahko skrajšamo tako, da namesto označevanja posameznih slikovnih elementov, le približno označimo robove cestišča. Takim oznakam pogosto pravimo grobe oznake, saj so običajno manj natančne, vendar pa jih v istem času lahko pridobimo veliko več.

1.3 Sorodna dela

Z željo po hitrejšem napredku na področju avtonomnih vozil se je v preteklih letih pojavilo veliko zbirk namenjenim segmentaciji urbanih območij [2] [4]. Ker je ročno označevanje slik časovno zahtevno, se veliko pozornosti namenja ustvarjanju sintetičnih zbirk, kjer slike in oznake proizvede kar računalnik [5]. Drugi pristopi so se osredotočili na uporabo obstoječih zbirk namenjenim klasifikaciji [6]. Gre za tako imenovano oddaljeno nadzorovano učenje (angl. distantly supervised learning), kjer avtorji članka za generiranje šibkih segmentacijskih mask cestišča uporabljajo obstoječe zbirke slik drugih domen. Šibke oznake nato kompenzirajo z večjim številom slik, ki jih uporabijo za iterativno učenje popolnoma konvolucijske nevronske mreže.

Prav tako je bilo v preteklosti opravljenih že veliko raziskav na področju semantične segmentacije cestišča. Članek [7] predlaga izboljšavo segmentacije z vpeljavo geometrijskih zakonitosti, pridobljenih s pomočjo statističnih metod, kot je mešanica Gaussovih modelov (angl. Gaussian Mixture Model). Uporaba dodatne informacije o položaju cestišča, kot je predstavljeno

v članku, prinese izboljšave predvsem na slikah zasneženih vozišč, kjer robovi ceste niso dobro vidni.

Ker problem semantične segmentacije ni omejen le na področje avtonomnih vozil, večina znanstvenih raziskav v povezavi z arhitekturami segmentacijskih mrež izhaja iz drugih domen. V grobem bi lahko razdelili segmentacijske nevronske mreže v dve skupini. V prvo uvrščamo t.i. nevronske mreže arhitekture kodirnik-dekodirnik. Njihova glavna značilnost je, da so sestavljene iz kodirnika, ki ga najpogosteje predstavlja kar klasifikacijska mreža kot je ResNet-50 [8], InceptionV3 [9], DenseNet [10], ipd. in dekodirnika, ki je odgovoren za zviševanje ločljivosti in končno napovedovanje kategorij posameznih slikovnih elementov. Primeri take arhitekture so nevronske mreže FCN [11], RefineNet [12] in U-Net [13], katero si bomo v nadaljevanju tudi podrobneje ogledali.

V drugo skupino uvrščamo nevronske mreže, ki nimajo kodirnika in se ločljivost posledično ne znižuje z globino mreže. Namesto nivojev združevanja običajno uporabljajo razširjeno konvolucijo (angl. dilated convolution) [14], s katero si pomagajo zajeti širši globalni kontekst. Zviševanje ločljivosti v tem primeru ni potreben, saj so dimenzije izhodov skritih nivojev pa tudi izhodnega nivoja ne spreminjajo. V praksi nevronske mreže te arhitekture dosegajo nekoliko boljše rezultate, vendar pa so računsko pogosto nekoliko bolj zahtevne. Najbolj znana primera mrež take arhitekture sta DeepLabV3+ [15] in PSPNet [16].

Zaradi pomanjkanja večjih zbirk za detekcijo cestnih poškodb v preteklosti še ni bilo veliko pristopov k reševanju tega problema s pomočjo konvolucijskih mrež. Delo [17] predstavlja tak pristop, kjer avtorji dela poročajo o omejitvah konvolucijskih mrež, saj pogosto zamenjujejo sence z razpokami vozišča in luže z luknjami na cestišču. A kljub omejitvam avtorji v delu s konvolucijskimi mrežami dosežejo 98% natančnost napovedi, kar je 3% višja natančnost od tradicionalnih pristopov, ki ne uporabljajo globokega učenja [18].

1.4 Prispevki

Detekcija poškodb cestišč omogoča analizo, ki daje vpogled v obrabljenost cestnih odsekov in vzdrževalcem nudi nekoliko bolj objektivno primerjavo poškodovanosti različnih odsekov ter vpogled nad tem, kateri deli ceste so bolj poškodovani, kako velike in kakšne so te poškodbe. Hkrati pa taki sistemi dajejo vpogled nad hitrostjo obrabe posameznih cestišč in pomagajo pri bolj učinkovitem načrtovanju obnove poškodovanih odsekov.

Naša želja je ustvariti algoritem, ki zna iz slike cestišča oceniti njegovo kvaliteto s pomočjo metrike MSI. Ker pa slikovne zbirke cestišč, označenih s pripadajočim indeksom MSI, žal niso javno dostopne, bomo naš problem nekoliko poenostavili in namesto indeksa MSI napovedovali tip površine cestišča. Ta bo predstavljal eno od treh vrednosti - makadamske cestišče, asfaltirano cestišče ali pa asfaltirano cestišče s talnimi oznakami oziroma talno signalizacijo.

V nalogi bomo pokazali, kako k takemu problemu pristopiti z uporabo nevronskih mrež. Podrobneje si bomo pogledali, kako lahko uporabimo konvolucijske nevronske mreže za segmentacijo cestišča in klasifikacijo tipa površine vozišča. Hkrati bomo pokazali tudi, kako lahko uporabimo segmentacijo za izboljšavo klasifikacije.

Glavni znanstveni prispevek bo raziskava mehanizmov za usmerjanje pozornosti nevronskih mrež v času učenja. V primeru segmentacije bomo pogledali, kako lahko določenim delom slike dodelimo različno težo in s tem poizkusimo izboljšati natančnost prepoznave cestišča. Klasifikacijo tipa površine vozišča pa bomo poizkusili izboljšati z vpeljavo informacije o položaju cestišča, ki jo pridobimo s pomočjo segmentacijske nevronske mreže.

1.5 Pregled poglavij

Naloga je razdeljena na šest poglavij. V naslednjem poglavju se bomo poglobili v osnovne koncepte navadnih in konvolucijskih nevronskih mrež ter njihovo učenje. V tretjem poglavju bomo obravnavali klasifikacijske mreže in

mreže namenjene segmentaciji slik, prav tako pa bomo predstavili predlagane izboljšave. V četrtem poglavju se bomo seznanili s podatki, ki jih imamo na razpolago za segmentacijo in s podatki za klasifikacijo ter si ogledali metode, s katerimi bomo ocenjevali uspešnost naših rešitev, hkrati pa nam bodo v pomoč pri primerjavi različnih modelov. Eksperimentalne rezultate bomo zbrali v petem poglavju. V zadnjem poglavju pa bomo opisali še sklepne ugotovitve, ki smo jih spoznali skozi celotno delo.

Poglavje 2

Umetne nevronske mreže

V tem poglavju si bomo pogledali, kaj umetne nevronske mreže sploh so, iz kje izhajajo in zakaj so v zadnjih letih postale tako popularne.

Pristopa, ki ju bomo uporabili za segmentacijo ceste in klasifikacijo tipa njene površine spadata v skupino nadzorovanega učenja (angl. supervised learning). Nadzorovano učenje je področje strojnega učenja, ki se v glavnem ukvarja z napovedovanjem. Bistvo nadzorovanega učenja je dostopnost in uporaba izhodnih podatkov oziroma napovedi, ki jim pogosto pravimo tudi temeljna resnica (angl. ground truth). Cilj nadzorovanega učenja je poiskati funkcijo, ki bo vhodne podatke učinkovito preslikala v izhodne podatke.

V našem primeru bomo to funkcijo predstavili z umetno nevronsko mrežo. Proces iskanja ustreznih parametrov funkcije pa imenujemo učenje nevronske mreže, za katerega potrebujemo podatkovno zbirko, ki jo sestavljajo pari vhodnih in izhodnih podatkov.

Kot pove že samo ime, inspiracija za umetne nevronske mreže izvira iz možganov. Tako kot možgani je tudi umetna nevronska mreža sestavljena iz mnogih nevronov, ki so med seboj povezani s povezavami, namenjenimi komuniciranju in na ta način tvorijo mrežo. Kljub analogiji pa je potrebno poudariti, da so umetne nevronske mreže zgolj poenostavitev katerihkoli možganov, še posebej človeških.

2.1 Arhitektura

2.1.1 Umetni nevron

Nevroni so torej osnovni gradniki umetne nevronske mreže. Imajo več različno uteženih vhodnih povezav, katerih uteženo vsoto s pomočjo t.i. aktivacijske funkcije (angl. activation function) preslikajo v izhod nevrona.

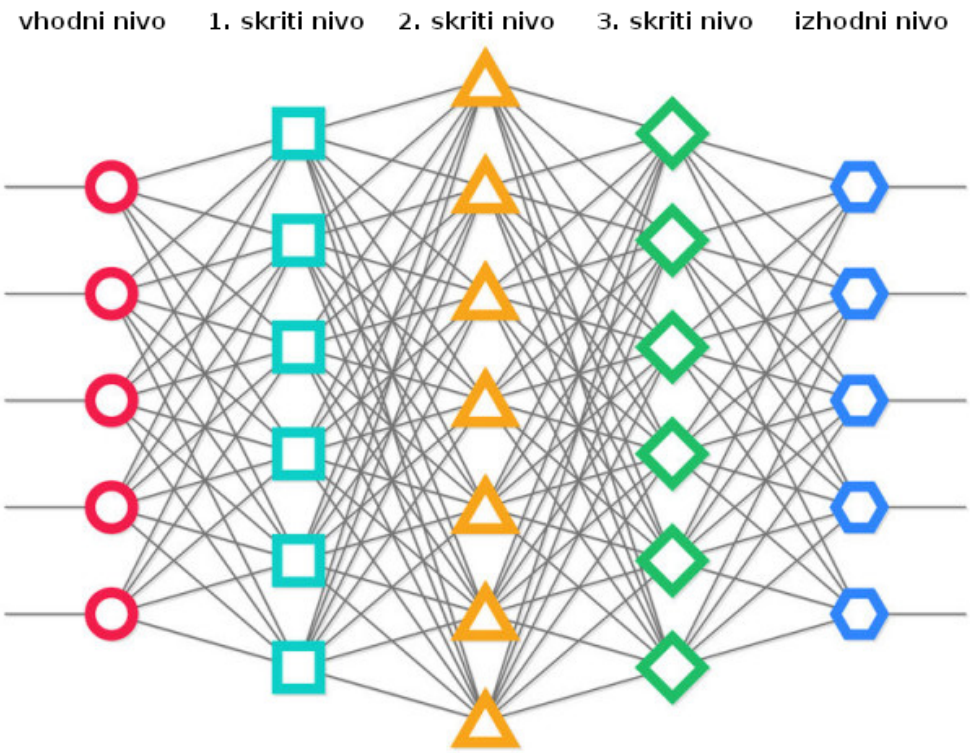
$$y = f\left(\sum_i x_i w_i + b\right) \tag{2.1}$$

Enačba (2.1) prikazuje matematičen zapis aktivacijske funkcije. Gre za parametrično funkcijo, kjer x_i predstavlja vhodni podatek, y izhod oziroma odziv nevrona na vhod, w_i in b pa parametre funkcije oziroma uteži, ki predstavljajo lastnosti in notranje stanje nevrona. Kakšna naj bo preslikava iz utežene vsote vhodov v izhod, je v veliki meri odvisno od problema, ki ga želimo rešiti. V praksi se za aktivacijske funkcije običajno uporablja nelinearne funkcije, kot so sigmoidna funkcija, hiperbolični tangens ali pa ReLU (angl. Rectified Linear Unit), ki je definiran z enačbo

$$f(z) = max(0, z). \tag{2.2}$$

2.1.2 Povezovanje v mrežo

Povezovanje nevronov dosežemo tako, da izhod nekega nevrona oz. različnih nevronov uporabimo kot vhod v naslednjega. Tako lahko prepletamo mnogo nevronov na zelo različne načine. Prepletanje lahko vsebuje cikle ali pa tudi ne, a najpogosteje se srečujemo z naprej-povezanimi nevronskimi mrežami (angl. Feedforward Neural Network). V takih mrežah lahko nevrone razvrstimo v nivoje, kjer so globlji nivoji odvisni le od višjih nivojev. Vsaka nevronska mreža ima vhodni in izhodni nivo, ter poljubno mnogo t.i. skritih nivojev (angl. hidden layers). Slika 2.1 prikazuje primer naprej-povezane zgradbe nevronske mreže s tremi skritimi nivoji. Vsi nivoji so polno-povezani



Slika 2.1: Prikaz ene izmed možnih naprej-povezanih arhitektur nevronske mreže s petimi vhodnimi in petimi izhodnimi nevroni ter tremi skritimi nivoji.

(angl. fully connected layers), kar pomeni, da so vsi nevroni na nekem nivoju povezani z vsemi nevroni na naslednjem nivoju.

Tako povezovanje osnovnih gradnikov spominja na zlaganje otroških Lego kock, kjer zgornjo stranico vsake kocke lahko uporabimo za nadaljnjo gradnjo. Koliko in katere gradnike bomo uporabili, kako jih bomo sestavili v celoto in kako visoka oziroma globoka (pri nevronskih mrežah namreč običajno govorimo o globini nivojev) bo končna struktura, pa je odvisno od razvijalca ter predvsem od problema, ki ga želimo rešiti.

2.2 Tok podatkov

Vhodni nivo je odgovoren za sprejem vhodnih podatkov, zato morajo biti ti predstavljeni na primeren način. Vsak nevron na podlagi pripadajočih vhodnih podatkov izračuna uteženo povprečje njegovih vhodov in s pomočjo aktivacijske funkcije izračuna svoj odziv, ki predstavlja njegov izhod. Na Sliki 2.1 lahko vidimo, da vhodne podatke nevrona na prvem skritem nivoju predstavljajo vsi izhodi nevronov na vhodnem nivoju. To pomeni, da je za izračun aktivacije nevrona na drugem skritem nivoju potrebno predhodno izračunati vse izhode nevronov na prvem skritem nivoju. Tako se postopek računanja izhodov in njihovega širjenja odvija vse do izhodnega nivoja (na Sliki 2.1 torej iz leve proti desni), proces pa imenujemo razširjanje naprej (angl. forward pass ali forward propagation).

Število izhodnih nevronov je odvisno od problema, ki ga rešujemo oziroma predstavitve izhodnih podatkov. V kolikor uporabljamo razredno predstavitev z bitnim kodiranjem, imamo za vsak izhodni podatek K izhodnih nevronov, kjer K predstavlja število razredov. V primeru klasifikacije imamo le en izhodni podatek - razred, ki predstavlja tip cestišča. Za potrebe segmentacije pa imamo toliko izhodnih podatkov, kot je ločljivost vhodne slike, zato ima izhodni nivo $višina\_slike \times širina\_slike \times K$ nevronov.

2.3 Vzvratno razširjanje

Tako kot večina algoritmov strojnega učenja tudi nevronske mreže za uspešno delovanje potrebujejo predhodno učenje. Učenje poteka tako, da na začetku naključno inicializiramo vse parametre nevronske mreže, nato pa čez mrežo spustimo vse učne primere in vrednosti na izhodu mreže primerjamo s temeljnimi resnicami.

Za primerjavo napovedi in temeljnih resnic definiramo funkcijo izgube (angl. loss function), katere lastnost je, da zavzema nizko vrednost, kadar so si temeljne resnice in napovedi podobne, oziroma visoko vrednost, kadar so si različne. Enostaven primer funkcije izgube je vsota kvadratov razlik ali SSD

(angl. Sum of Squared Difference). Za naše potrebe pa bomo za funkcijo izgube uporabili kategorično prečno entropijo (angl. categorical cross entropy). Kategorično prečno entropijo matematično zapišemo z enačbo (3.3), kjer w predstavlja parametre oziroma uteži nevronske mreže, N število primerov v učni množici, K število klasifikacijskih razredov, $y^{(k)}_{true}$ temeljno resnico k-tega učnega primera in $y^{(k)}_{predicted}$ napoved nevronske mreže za k-ti učni primer.

$$C(\mathbf{w}) = -\sum_{i=1}^{N}\sum_{k=1}^{K} y^{(k)}_{true} * \log(y^{(k)}_{predicted}) \tag{2.3}$$

Tu je potrebno poudariti, da smo izraz učni primer nekoliko izrabili, saj ima različen pomen glede na to ali govorimo o klasifikaciji ali segmentaciji. Enačba (3.3) nekoliko bolje opisuje funkcijo izgube, ki jo pogosto uporabljamo pri učenju klasifikacijskih mrež, vendar pa se od funkcije, ki bi jo uporabili v primeru segmentacije, ne razlikuje bistveno. Ker pri segmentaciji učni primer predstavlja slikovni element, se na sliki nahaja toliko učnih primerov, kot je ločljivost slike. Enačbo (3.3) za potrebe segmentacije zato priredimo tako, da $y^{(k)}_{true}$ in $y^{(k)}_{predicted}$ ne predstavljata več napake na ravni slike temveč na ravni slikovnih elementov. Tako v času učenja seštevamo napake posameznih slikovnih elementov oziroma segmentacijski učnih primerov.

Matematično gledano učenje predstavlja iskanje takšnih parametrov nevronske mreže, da je povprečna vrednost funkcije izgube vseh učnih primerov minimalna. Gre seveda za ničelno vrednost odvoda funkcije izgube, kar opisuje enačba (2.4):

$$\frac{\partial C}{\partial \mathbf{w}}(\mathbf{w}) = 0 \tag{2.4}$$

Za zapletene funkcije, kot je izhod nevronske mreže, je odvod analitično nemogoče poiskati, zato si v praksi pomagamo z iterativnimi optimizacijskimi algoritmi. Verjetno najbolj prepoznaven primer takega algoritma je gradientni sestop (angl. gradient descent ali steepest descent). Danes pa se v praksi uporabljajo tudi izboljšave te metode, kot je Adam (krajše za Adaptive Moment Estimation), katerega pri učenju nevronske mreže uporabljamo

tudi mi. Oba algoritma v osnovi delujeta tako, da izračunata negativni gradient funkcije izgube, ki nam pove, kako je potrebno spremeniti parametre mreže, da zmanjšamo vrednost funkcije izgube.

Ker so nevroni na nižjih nivojih odvisni od nevronov na višjih nivojih, je za spremembo parametrov mreže potrebno izračunati, kakšen vpliv na vrednost funkcije izgube ima posamezen parameter. To storimo tako, da z uporabo verižnega pravila za odvajanje izračunamo lokalne gradiente aktivacijskih funkcij, ki nam povedo, kako spremeniti njihove vhode. Tej tehniki propagiranja napake običajno pravimo vzvratno razširjanje, saj napako pridobljeno na izhodnem nivoju širimo v obratni smeri toka podatkov nazaj po mreži (na sliki 2.1 od desne proti levi).

Učenje nevronske mreže torej poteka v treh korakih. V prvem koraku čez mrežo spustimo vse učne primere in na izhodih seštevamo napake. V drugem koraku v obratni smeri propagiramo napake in izračunamo lokalne gradiente aktivacijskih funkcij. Nazadnje na podlagi lokalnih gradientov posodobimo parametre nevronske mreže. Postopek ponavljamo vse do konvergence, ki jo določimo na podlagi napake, narejene na validacijski množici (angl. validation set). Ločeno množico za določanje konvergence potrebujemo zato, ker je ocenjevanje na učni množici pristransko in lahko učenje vodi v prekomerno prilagajanje učni množici (angl. overfitting). Pogosta lastnost prekomernega prilagajanja je, da parametri mreže zavzemajo visoke vrednosti. Proti prekomernem prilagajanju se borimo tako, da funkciji izgube dodamo člen za regularizacijo parametrov (angl. weight regularization), zapišemo pa jo lahko na sledeči način:

$$C_{total} = C(f(\mathbf{x}), \mathbf{W}) + \lambda R(\mathbf{W}), \tag{2.5}$$

kjer parameter λ predstavlja stopnjo regularizacije in določa, kolikšen vpliv naj ima sama regularizacija, funkcija R pa določa, na kakšen način vrednotimo skupno velikost parametrov mreže. V praksi za funkcijo R običajno uporabljamo kar normo L1 ali L2, stopnjo regularizacije pa obravnavamo kot dodatni hiper-parameter učenja, katero vrednost običajno določimo eksperi-

mentalno.

2.4 Konvolucijske nevronske mreže

Sedaj, ko razumemo osnovno delovanje in potek učenja nevronskih mrež, si poglejmo, v čem se razlikujejo konvolucijske nevronske mreže (angl. Convolutional Neural Networks, v nadaljevanju tudi CNN) in zakaj je njihova uporaba tako priljubljena na področju računalniškega vida.

Glavni problem nevronskih mrež s polno-povezanimi nivoji je število učnih parametrov. Problem postane še posebej očiten pri globokih nevronskih mrežah in mrežah z veliko vhodnimi podatki. Ker je pri razpoznavi slikovne vsebine vhodnih podatkov toliko, kolikor je ločljivost vhodne slike, je uporaba polno-povezanih nivojev v veliki meri neprimerna. Rešitev je uporaba konvolucijskih nivojev, ki namesto povezovanja vseh nevronov nekega nivoja, povezuje nevrone, ki se nahajajo znotraj nekega omejenega območja. To zmanjšuje število učnih parametrov nevronske mreže, hkrati pa ohranja predpostavko, da so bližnji slikovni elementi vsebinsko bolj povezani, kot tisti bolj oddaljeni.

Poglejmo si zato podrobneje, kaj pravzaprav je konvolucija in kako jo uporabimo za gradnjo nevronskih mrež.

2.4.1 Konvolucijski nivo

Tako kot običajne nevronske mreže so tudi CNN sestavljene iz posameznih nivojev. V domeni slikovnega vhoda je vsak od nivojev konvolucijske mreže odgovoren za prevzem tri-dimenzionalnega tenzorja in njegovo preslikavo v nov tri-dimenzionalni izhodni tenzor s pomočjo odvedljive aktivacijske funkcije.

Najpomembnejši gradnik CNN je konvolucijski nivo, ki je hkrati tudi prvi nivo konvolucijske nevronske mreže. Vsak konvolucijski nivo vsebuje enega ali več konvolucijskih filtrov oziroma jeder, s katerimi izračuna izhode lokalno povezanih nevronov. Konvolucijski filtri pravzaprav predstavljajo

parametre nevronske mreže, kar pomeni, da se jih v fazi učenja optimiziramo s pomočjo gradientnega sestopa in vzvratnega razširjanja, kot smo si to ogledali v prejšnjem poglavju. Filtri so v praksi najpogosteje majhni in imajo običajno višino enako širini.

Konvolucijo si lahko predstavljamo kot drsenje filtra čez vhodni tenzor, matematično pa jo lahko opišemo s formulo:

$$(I * f)(x, y) = \sum_i \sum_j I(i, j) f(x - i, y - j), \tag{2.6}$$

kjer f predstavlja filter in I vhodni tenzor, ki na prvem nivoju predstavlja kar vhodno sliko. Višina oziroma širina filtra sta običajno manjši ali pa kvečjemu enaki višini in širini vhodnega tenzorja, medtem ko je globina filtra vedno enaka globini vhodnega tenzorja. V nadaljevanju bomo obravnavali simetrično konvolucijo, kar pomeni, da sta širina in višina jedra enaki, uporabljali pa bomo raje izraz velikost jedra. Podobno velja tudi za vhodni in izhodni tenzor, saj v praksi pogosto vhodno sliko predhodno preoblikujemo v kvadratno obliko.

V kolikor je velikost filtra večja od ena, je izhodni tenzor konvolucije manjši od vhodnega. V kolikor si želimo ohraniti dimenzijo vhodnega tenzorja, moramo vhodni tenzor pred samo operacijo konvolucije razširiti. Zaradi preprostosti za razširitev (angl. padding) najpogosteje uporabimo kar vrednost nič, čeprav obstajajo tudi alternativne možnosti, kot na primer kopiranje najbližjih elementov.

Samo drsenje jedra je možno izvajati na več načinov. Osnovna možnost je premikanje s korakom (angl. stride) enakim ena, kar pomeni, da filter položimo na vsa možna mesta vhodnega tenzorja. Korak pa je lahko tudi več kot ena, kar pomeni, da določena mesta vhodnega tenzorja preskočimo. Vrednost koraka je enden izmed hiper-parametrov konvolucijskega nivoja, katerega nastavimo pred učenjem in se ga ne učimo.

Velikost izhodnega tenzorja je odvisna od velikosti vhodnega tenzorja, razširitve pred samo operacijo, velikosti jedra in koraka konvolucije. Izračunamo

pa jo lahko po naslednji formuli:

$$O = \frac{N + 2P - F}{S} + 1, \tag{2.7}$$

kjer je O velikost izhoda, N velikost vhodnega tenzorja, F velikost filtra, P razširitev, S pa korak s katerim premikamo jedro.

Kot že omenjeno na nekem konvolucijskem nivoju običajno uporabimo več filtrov, kar tudi definira samo globino izhodnega tenzorja. Izhodi konvolucij z vsemi filtri na nekem nivoju imajo namreč vedno enako višino in širino, zato jih lahko zložimo skupaj in iz njih sestavimo nov tri-dimenzionalni izhodni tenzor. Le-ta pa predstavlja vhod v naslednji nivo, ki je najpogosteje kar nivo paketne normalizacije [19] (angl. Batch Normalization), sledi pa mu nivo nelinearnosti (običajno je to enota ReLU, ki smo jo spoznali že v predhodnem poglavju).

Izhodnim tenzorjem po nivoju nelinearnosti pogosto pravimo tudi aktivacijske maske (angl. activation map) ali maske značilk (angl. feature map). Z učenjem konvolucijske mreže namreč prilagajamo filtre konvolucijskih nivojev, ki se spreminjajo na tak način, da rezultat konvolucije med filtrom in vhodnim tenzorjem zavzema visoko vrednost, na mestih kjer je filter podoben vhodnemu tenzorju.

Vizualiziranje odzivov slik na posamezne filtre predhodno naučene konvolucijske mreže pokaže, da se filtri nižjih konvolucijskih nivojev učijo enostavnih značilk, kot so robovi, črte ali barvne spremembe. Ker so višji konvolucijski nivoji odvisni od izhodov nižjih nivojev, se posledično učijo kompleksnejših slikovnih značilnosti, kot so krivulje, vzorci, itd. Na konvolucijske filtre zato lahko gledamo kot na nekakšne predloge značilk, ki jih CNN uporabi za ekstrakcijo večdimenzionalne maske značilk pripadajoče vhodne slike.

2.4.2 Nivo združevanja

Poleg konvolucijskega nivoja in nivoja nelinearnosti v CNN najpogosteje srečamo še nivo združevanja (angl. Pooling layer). Nivo združevanja za

razliko od konvolucijskega ne vsebuje filtrov in posledično nima učnih parametrov, njegova naloga pa je zmanjšati prostorsko dimenzijo maske značilk. S tem dosežemo, da se zmanjša število učnih parametrov, hkrati pa se zmanjša tudi lokacijska odvisnost značilk naučenih na nekem višjem nivoju.

Tudi operacijo združevanja si lahko predstavljamo kot drsno okno, kjer je običajno premični korak enak velikosti okna, okno pa drsi po vsaki globinski rezini posebej. Izhod na danem položaju je največkrat kar maksimalna vrednost vhodnega tenzorja znotraj okna, v tem primeru govorimo o maksimalnem združevanju (angl. Max Pooling). Matematično bi operacijo zapisali na sledeč način:

$$y(x', y', z) = max(I(x_{i:i+s}, y_{j:j+s}, z)), \tag{2.8}$$

kjer I predstavlja vhodni tenzor, s pa velikost okna in premični korak. Spremenljivki x' in y' predstavljata lokacijo na izhodnem tenzorju manjših dimenzij.

Alternativno bi lahko namesto maksimalne vrednosti uporabili tudi kakšen drugačen izračun, na primer povprečje (angl. Average Pooling) elementov znotraj okna ali pa normo L2 (angl. L2 norm Pooling), a to v praksi redko zasledimo.

2.4.3 Polno-povezani nivo

Polno-povezani nivo (angl. Fully Connected Layer) smo sicer že spoznali. Gre za nivo, kjer so vsi nevroni povezani z vsemi nevroni naslednjega nivoja. Prav tako smo se seznanili, da uporaba polno-povezanih nivojev pogosto ni primerna, saj vsebuje veliko število parametrov. Čeprav to v veliki meri drži, polno-povezane nivoje v konvolucijskih mrežah kljub temu zasledimo. Običajno se nahajajo globoko v mreži, kjer je resolucija maske značilk že dovolj nizka. Delu nevronske mreže, ki se nahaja višje od prvega polno-povezanega nivoja, pogosto pravimo kodirnik ali ekstraktor značilk (angl. encoder ali feature extractor), saj iz prvotnega vhoda izlušči le relevantne značilnosti. Te značilnosti v primeru klasifikacije nato uporabimo za vhod v

polno-povezani nivo, ki je namenjen uporabi naučenih značilk za napovedovanje tarčne spremenljivke.

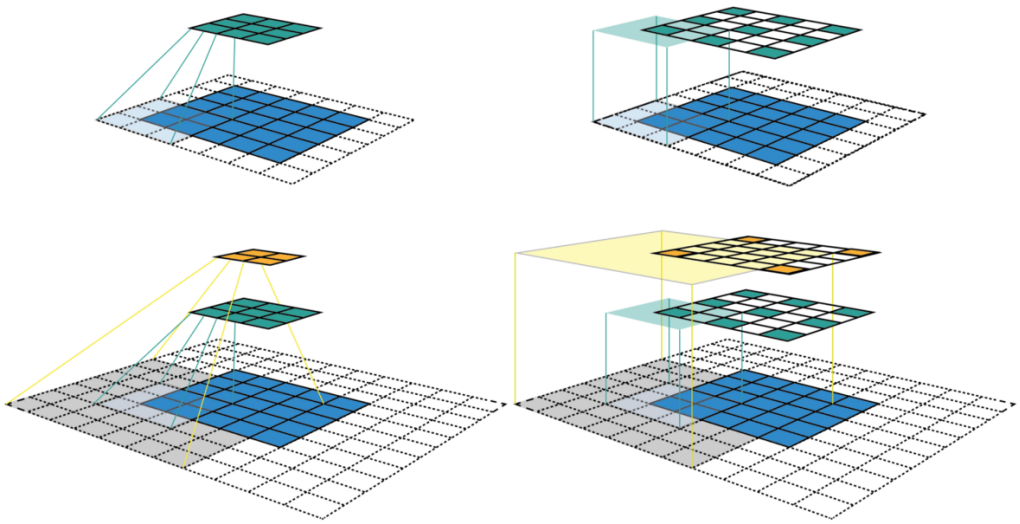
V segmentacijskih nevronskih mrežah pa polno-povezanih nivojev ponavadi ne srečamo, saj kodirniku običajno sledi dekodirnik (angl. decoder), kot bomo to spoznali v prihodnjih poglavjih. Na tem mestu omenimo le, da je naloga dekodirnika zviševanje resolucije maske značilk, pridobljene s pomočjo kodirnika.

2.4.4 Popolnoma konvolucijska mreža

Do nedavnega je bila uporaba polno-povezanih nivojev na koncu nevronske mreže nekaj običajnega. Leta 2014 se je to spremenilo, ko je bila predstavljena t.i. popolnoma konvolucijska mreža (angl. Fully Convolutional Network) ali krajše FCN. Glavni prispevek članka [20] je ugotovitev, da lahko polno-povezani nivo implementiramo s pomočjo konvolucijskega nivoja.

Opazimo lahko, da je edina razlika med polno-povezanim in konvolucijskim nivojem v tem, da so nevroni konvolucijskega nivoja med seboj povezani le v lokalni regiji velikosti filtra. Sama funkcionalnost obeh nivojev pa je ista, saj lahko obe operaciji matematično predstavimo kot vsoto produktov istoležnih elementov matrik. Posledično lahko polno-povezane nivoje nadomestimo s konvolucijskimi tako, da pri konvoluciji uporabimo filter enakih dimenzij, kot je dimenzija vhodnega tenzorja.

Glavna razlika med tradicionalnimi konvolucijskimi mrežami s polno-povezanimi nivoji in FCN je v tem, da je izhod FCN maska značilk in ne le klasifikacijski rezultat za posamezen razred. To ima izjemen pomen za segmentacijo, saj značilke izhodne maske FCN ohranjajo lokacijsko informacijo o značilkah predhodnega nivoja na katere se posamezna značilka izhodne maske odziva. To je v neposredni relaciji s tako imenovanim sprejemnim poljem (angl. receptive field), katerega razumevanje igra ključno vlogo pri načrtovanju segmentacijske mreže in si ga zato poglejmo v naslednjem poglavju.



Slika 2.2: Prikaz dveh različnih načinov vizualizacije mask značilk (vsak v svojem stolpcu). V levem stolpcu lahko vidimo običajni prikaz mask značilk dveh (v prvi vrsti) oziroma treh nivojev (v drugi vrsti). Iz slik v levem stolpcu je sprejemno polje težko razbrati, zato za prikaz sprejemnega polja velikost mask značilk raje fiksiramo, kot je to prikazano v desnem stolpcu [21].

2.4.5 Sprejemno polje

Sprejemno polje (angl. receptive field) je definirano kot del vhodnega tenzorja, ki vpliva na neko značilko na višjem nivoju konvolucijske mreže. Vhodni tenzor lahko predstavlja vhodna slika ali pa izhod nekega nižjega nivoja, zato sprejemno polje vedno računamo relativno na nek določen nivo. A v praksi nas najpogosteje zanima, kakšno je sprejemno polje na izhodu nevronske mreže glede na vhodno sliko.

Slika 2.2 prikazuje uporabo konvolucije z jedrom velikosti 3×3 in korakom 2 na vhodnem tenzorju 5×5, ki ga pred operacijo razširimo za ena. Velikost izhodnega tenzorja je enaka 3×3, velikost sprejemnega polja nevronov po operaciji pa je enaka velikosti jedra, kar je na Sliki 2.2 prikazano v prvi vrstici. Če konvolucijo z istimi lastnosti ponovimo na izhodnem tenzorju,

dobimo nov tenzor velikost 2×2, velikost efektivnega sprejemnega polja pa se poveča na 7×7 - prikazano v drugi vrstici Slike 2.2.

Iz tega lahko razberemo, da sta velikost efektivnega sprejemnega polja in velikost izhodnega tenzorja v obratnem sorazmerju. Z zmanjševanjem dimenzije izhodnih tenzorjev povečujemo sprejemno polje, kar v konvolucijskih mrežah pogosto dosežemo z uporabo nivojev združevanja. Zmanjševanje ločljivosti pa ni edini pristop k povečevanju sprejemnega polja. Tega lahko lahko povečamo tudi s t.i. razširjenimi konvolucijami (angl. dilated convolution), ki si jih bomo podrobneje ogledali v naslednjem poglavju.

Veliko sprejemno polje si želimo predvsem zaradi globalnega konteksta. Kot že vemo iz preteklega poglavja, s konvolucijskimi nivoji odkrivamo lokalne značilnosti vhodnih tenzorjev. Razširitev sprejemnega polja pa nam omogoča zajemanje informacije večje prikazanih objektov [22], s tem pa odkrivanje globalnega konteksta, ki je ključnega pomena za računalniško razumevanje slikovne vsebine.

Poglavje 3

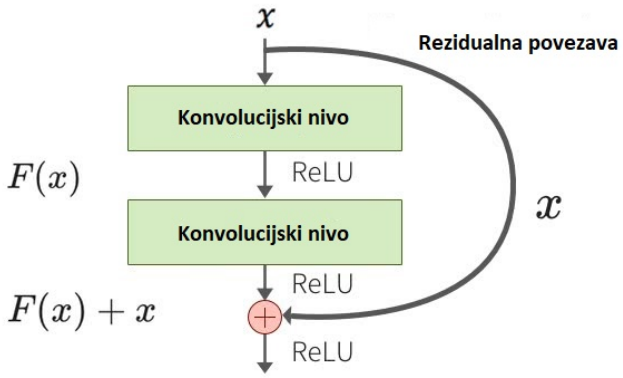
Klasifikacija in segmentacija

3.1 Klasifikacija površine

Sedaj, ko razumemo delovanje konvolucijskih nevronskih mrež in poznamo proces njihovega učenja, si poglejmo, kako lahko uporabimo obstoječe nevronske mreže, ki so se zaradi svoje arhitekture v preteklih letih izkazale za zelo učinkovite, na problemih, ki so podobni našemu.

3.1.1 ResNet-50

Analiza naučenih nevronskih mrež je pokazala, da se konvolucijski filtri na začetnih konvolucijskih nivojih učijo preprostih značilk, kot so robovi, preproste barve, krivulje, ipd. Za prepoznavanje kompleksnejših slikovnih lastnosti pa potrebujemo več konvolucijskih nivojev, zato se je v preteklih letih uveljavil trend globokih nevronskih mrež (angl. Deep neural networks). Globoke nevronske mreže so v praksi pokazale izjemen napredek z vidika klasifikacijske točnosti, vendar pa je postopek učenja hkrati postal težji. Učenje globokih nevronskih mrež je zapletenejše predvsem zaradi pojava, ki mu pravimo problem izginjajočega gradienta (angl. vanishing gradient problem). Problem nastopi, ko v fazi vzvratnega razširjanja gradient propagiramo proti nižjim nivojem. Vrednost gradienta se tako zaradi večkratnega množenja lahko približa ničelni vrednosti, kar pa onemogoča učenje.



Slika 3.1: Prikaz rezidualne povezave, kot je bila prvič predstavljena v arhitekturi mreže ResNet.

Reševanje problema je naslovila ekipa Microsoftovih raziskovalcev, ki je leta 2015 predstavila družino t.i. rezidualnih konvolucijskih nevronskih mrež (angl. Residual Networks), ali krajše ResNet[23]. Ključni napredek, ki ga je predstavila raziskava, je vpeljava rezidualnih povezav (angl. residual connection, pogosto tudi identity shortcut connection), ki služijo kot bližnjice med nivoji, ki med seboj sicer niso neposredno povezani. Uporaba rezidualnih povezav zato omogoča preprostejše in učinkovitejše učenje globjih nevronskih mrež.

V praktičnem delu naloge si bomo za potrebe klasificiranja tipa cestišča pomagali z mrežo arhitekture ResNet-50. Tudi za potrebo segmentacije cestišča bomo za kodirnik vzeli mrežo ResNet-50, ki pa jo bomo prilagodili tako, da bodo konvolucijski nivoji uporabljali razširjeno konvolucijo, kot je to opisano v članku [24]. K razširjenim konvolucijam se bomo še vrnili v prihodnjem poglavju, še prej pa si poglejmo, kakšne prednosti nam prinese uporaba popularne arhitekture kot je ResNet-50.

3.1.2 Prenos znanja

V prejšnjem poglavju smo omenili, da ob začetku učenja učne parametre nastavimo na naključno vrednost. To pogosto ni najprimernejša rešitev, saj

lahko zaradi naključne izbire konvergenca traja dolgo časa ali pa obtiči v lokalnem optimumu. Učenje nevronske mreže namreč lahko traja tudi več dni, v kolikor imamo veliko podatkovno zbirko z zelo raznolikimi podatki.

Prednost, ki si jo lahko privoščimo zaradi izbire mreže ResNet-50, je uporaba predhodno naučenih parametrov. Po dolgotrajnem učenju mrež priljubljenih arhitektur, znanstveniki za raziskovalne potrebe širši javnosti pogosto omogočijo dostop do njihovih modelov. Ti so ponavadi naučeni na splošnih zbirkah, kot sta ImageNet [25] ali MS COCO [26], učenje pa pogosto traja več dni. Kljub temu da je domena omenjenih zbirk običajno bistveno drugačna od ciljne domene, se uporaba predhodno naučenih uteži v praksi obrestuje.

V kolikor bi bili domeni podobni, bi lahko uporabili kar model, naučen na eni od omenjenih zbirk, ker pa imamo zelo specifično domeno, bomo uteži predhodno naučenih modelov uporabili le za inicializacijo naše nevronske mreže. Nato bomo nevronsko mrežo doučili s podatki naše domene, kot bi to tudi sicer storili v primeru naključne inicializacije učnih parametrov. Uporaba predhodno naučenih modelov je v praksi nekaj običajnega, metode pa se je prijel izraz *prenos znanja* (angl. transfer learning) [27].

Kot smo omenili, bomo v našem primeru za klasifikacijo tipa cestišča uporabili arhitekturo ResNet-50. Njene učne parametre bomo inicializirali s parametri, naučenimi na podatkovni zbirki ImageNet. Tu je potrebno izpostaviti, da ima zbirka ImageNet 1000 kategorij, kar pomeni, da ima predhodno naučena mreža ResNet-50 na izhodnem nivoju 1000 izhodnih nevronov, medtem ko naša arhitektura potrebuje le tri. Prenos znanja končnih polno-povezanih nivojev zato ne bo mogoč. Kljub temu da so značilke na globljih nivojih najbolj specifične za neko domeno, to ne predstavlja problema, saj nameravamo celotno mrežo doučiti s pomočjo naše podatkovne zbirke. Tako bomo učne parametre polno-povezanih nivojev inicializirali naključno, za vse predhodne nivoje pa bomo uporabili tehniko prenosa znanja.

3.2 Segmentacija ceste

V prejšnjem poglavju smo si pogledali vse operacije potrebne za sestavo preproste konvolucijske mreže. Omenili smo, da bi samo strukturo nevronske mreže lahko razdelili na dva dela. Prvi del v večini sestavljajo konvolucijski nivoji in nivoji združevanja. Večkrat ga imenujemo kodirnik (angl. encoder), saj nam omogoča ekstrakcijo slikovnih značilnosti, katere nato uporabi drugi del, pogosto imenovan klasifikacijska glava (angl. classification head), ki je odgovoren za napovedovanje (klasifikacijo ali regresijo). Semantična segmentacija gradi na temeljih, ki smo jih opisali v prejšnjih poglavjih, razširja pa mrežo z novim arhitekturnim delom, ki ga pogosto imenujemo dekodirnik (angl. decoder). Ta je umeščen med sam kodirnik in klasifikacijsko glavo, odgovoren pa je za obravnavo nizko ločljivostne maske značilk in povečanje velikosti na dimenzije enake vhodni sliki.

3.2.1 Zviševanje ločljivosti

Ko govorimo o skaliranju tenzorja (slike ali maske značilk), pravzaprav govorimo o spreminjanju njegove ločljivosti. Z zviševanjem ločljivosti tenzorju med obstoječe elemente vrivamo nove, katerih vrednost lahko izračunamo na različne načine.

Ena najenostavnejših metod za izračun vrednosti novih elementov je metoda najbližjih sosedov (angl. nearest neighbors). Metoda novemu elementu preprosto dodeli vrednost najbližjega elementa originalnega tenzorja. Zaradi svoje preprostosti je v izhodnem tenzorju možno opaziti neželene kockaste artefakte, zaradi česar se metoda v praksi pogosto ne uporablja.

Nekoliko bolj zanimivi pa sta metodi linearna interpolacija in transponirana konvolucija, ki ji večkrat pravimo tudi dekonvolucija (angl. transposed convolution, deconvolution ali pa tudi fractionally-strided convolution). Na tem mestu je potrebno poudariti, da izraz dekonvolucija morda ni najbolj primeren, saj v matematični terminologiji predstavlja obratno operacijo konvolucije, ki pa se od transponirane konvolucije razlikuje. Izraz se je na po-

dročju globokega učenja vseeno prijel, zato je prav, da ga omenimo, čeprav se ga bomo v nadaljevanju izogibali.

Obe metodi je v praksi mogoče srečati, zato si poglejmo, na kakšen način izračunata vrednosti novo vpeljanih slikovnih elementov.

Linearna interpolacija

Ker ločljivost kodirnikovega izhoda najpogosteje povečujemo v višino in širino hkrati, običajno govorimo o bi-linearni interpolaciji. Gre za dvokratno linearno interpolacijo, kjer v prvem koraku izračunamo vrednost točke v smeri x in nato še vrednost v smeri y.

Metoda predpostavlja, da so vrinjeni elementi v linearnem razmerju z najbližjimi elementi tenzorja osnovne ločljivosti. Vrednost novih elementov se tako izračuna kot uteženo povprečje najbližjih sosedov v dani smeri, kjer sta uteži odvisni od razdalje vrinjenega elementa do posameznega soseda.

Gre za enostaven in učinkovit izračun novih vrednosti, ki ne potrebuje nobenih učnih parametrov in zato v času učenja ne potrebuje dodatnega računalniškega spomina.

Transponirana konvolucija

Nekoliko drugačen pristop uporablja transponirana konvolucija. Cilj operacije je, empirično učenje optimalnega zviševanja ločljivosti. Za te potrebe uporablja učne parametre, katere tekom učenja optimiziramo s pomočjo vzvratnega razširjanja napake, tako kot vse ostale učne parametre nevronske mreže.

Kljub temu da se na prvi pogled rešitev zdi nekoliko bolj obetavna, se transponirana konvolucija v praksi pogosto ne uporablja. Glavni razlog za to je, da zviševanje ločljivosti s pomočjo transponirane konvolucije pogosto vpelje neželene artefakte, ki spominja na vzorec šahovnice (angl. checkerboard artifact) [28]. Prav tako z uporabo transponirane konvolucije dodamo mnogo učnih parametrov, kar terja večji računalniški pomnilnik, hkrati pa tudi več podatkov in daljše učenje.

V nadaljevanju zato ne bomo uporabljali transponirane konvolucije, temveč le preprosto bi-linearno interpolacijo, s katero v praksi lahko dosežemo zadovoljive rezultate.

3.2.2 Razširjena konvolucija

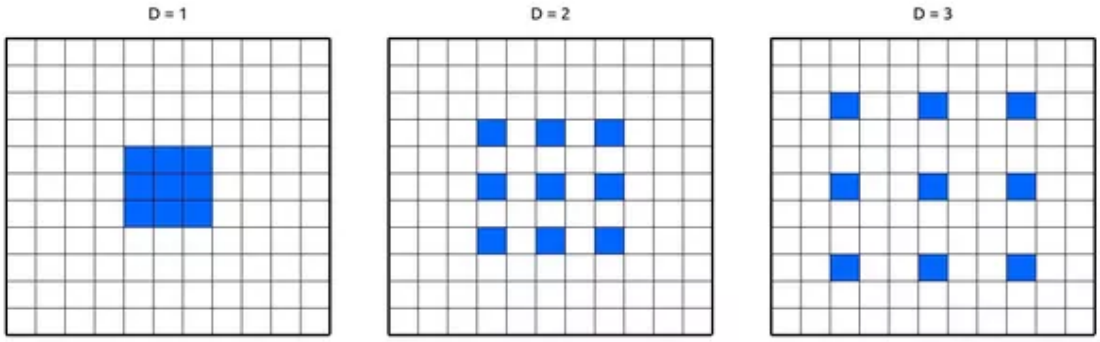
Poleg polno-povezanih nivojev, ki jih za potrebe segmentacije lahko nadomestimo s konvolucijskimi nivoji, kot smo to opisali v prejšnjem poglavju, eden od glavnih problemov ostaja uporaba nivojev združevanja. Nivoji združevanja nam omogočijo večje sprejemno polje, ki je pomembno za razumevanje globalnega slikovnega konteksta, a hkrati tudi zmanjša prostorsko ločljivost maske značilk in s tem do neke mere zavrže informacijo o lokaciji značilnosti, ki jo posamezna značilka opisuje.

Sprejemno polje lahko povečamo tudi z uporabo tako imenovane razširjene konvolucije (angl. dilated ali atrous convolution) [14], katere glavna prednost je, da preprečuje zmanjševanje izhodne ločljivosti. Osnovna ideja razširjene konvolucije je povečanje jedra z vstavljanjem praznih mest med sosednje element jedra. Gre za bolj generičen opis konvolucijske operacije z dodatnim parametrom imenovanim faktor razširitve (angl. dilation rate), ki opisuje razmak med posameznima elementoma jedra. Običajna konvolucija je torej le poseben primer razširjene konvolucije, kjer je faktor razširitve enak ena. Primer razširitve jedra je prikazan na Sliki 3.2, kjer skrajno leva skica prikazuje jedro običajne konvolucije, srednja in desna skica pa razširjeni jedri s faktorjem razširitve enakim dve in tri.

Enačbo (2.7) za izračun velikosti izhoda konvolucije tako lahko dopolnimo na sledeč način:

$$O = \frac{N + 2P - F - (F-1)(D-1)}{S} + 1, \tag{3.1}$$

kjer tako kot v enačbi (2.7) O predstavlja velikost izhoda, N velikost vhoda, F velikost filtra, P razširitev, S pa korak, s katerim premikamo jedro. Dodali pa smo le še parameter D, ki predstavlja faktor razširitve.

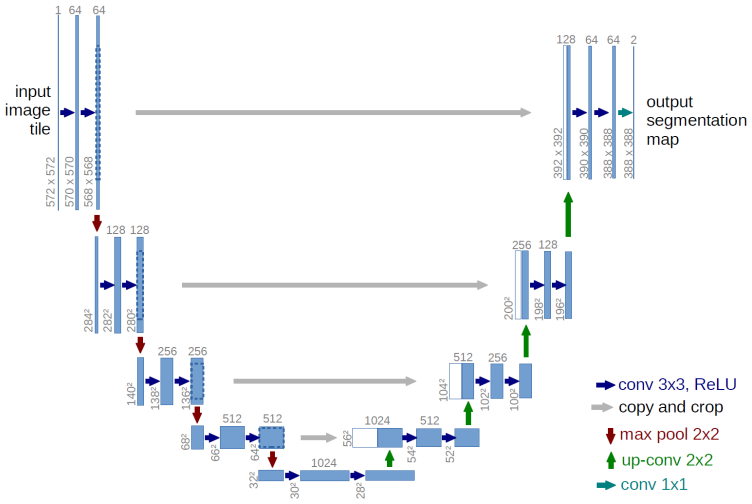


Slika 3.2: Prikaz treh različnih razširitev jedra. Od leve proti desni so jedra razširjena s faktorjem razširitve enakim ena (jedro, uporabljeno v tradicionalni konvoluciji), dve in tri.

3.2.3 U-Net

Za enega najpomembnejših prebojev z vidika natančnosti semantične segmentacije so poskrbeli avtorji članka [29]. Nevronska mreža U-Net uvaja novost združevanja (angl. concatenation) značilk na globljih nivojih s tistimi, naučenimi na višjih nivojih. Dekodirnik ima tako dostop do značilk, ki se nahajajo v kodirniku. Združevanje se običajno nahaja za nivojem zviševanja ločljivosti, kjer se izhod združi z značilkami iste ločljivosti določenega kodirnikovega nivoja. Arhitektura in združevanje značilk istih ločljivosti je prikazana na Sliki 3.3.

Za uporabo novosti, ki jo uvaja U-Net, ne potrebujemo posebne arhitekture kodirnika. V našem primeru bo vlogo kodirnika prevzela nevronska mreža ResNet-50. To nam omogoča uporabo prenosa znanja, kot smo to opisali v poglavju 3.1.2. Tu bomo za inicializacijo učnih parametrov (kjer je to mogoče) uporabili uteži, pridobljene s predhodnim učenjem na zbirki ImageNet, ostale pa bomo pred začetkom učenja nastavili na naključno vrednost. Mrežo bomo nato naučili na zbirki Cityscape [2], s pomočjo katere bomo opravili nekaj raziskovalnih eksperimentov, opisanih v naslednjem poglavju. Nato bomo učne parametre modela, s katerim smo dobili najboljše rezultate,



Slika 3.3: Prikaz arhitekture nevronske mreže U-Net, kjer modri pravokotniki predstavljajo konvolucijske nivoje s paketno normalizacijo in nivo nelinearnosti ReLU. Rdeče puščice predstavljajo operacijo zniževanja dimenzije, zelene operacije zviševanja dimenzije, sive pa operacijo združevanja [29].

uporabili za ponovno inicializacijo in doučili isto nevronsko mrežo na zbirki slik, podobnim tistim za klasifikacijo tipa cestišča.

3.3 Nadgradnje

V tem poglavju si bomo pogledali pristope s katerimi smo poiskusili izboljšati uspešnost klasifikacije in segmentacije ter zmanjšati razmak med uspešnostjo pri uporabi grobih anotacij v primerjavi z natančnimi.

3.3.1 Segmentacija

Obravnava robov cestišča

Obstoječe raziskave so pokazale, da imajo konvolucijske mreže na področju segmentacije cest pogosto težavo z razlikovanjem med cestiščem in pločnikom. To izzove vprašanje, ali je med učenjem možno usmeriti pozornost nevronske

mreže v robove cestišča.

Mehanizem za usmerjanje pozornosti je na področju semantične segmentacije nekoliko bolj intuitiven kot pri klasifikaciji, saj je rezultat segmentacije enakih dimenzij kot vhodna slika in vsak izhodni nevron predstavlja razred istoležnega vhodnega slikovnega elementa. Kot smo opisali v prejšnjem poglavju, v času učenja vodimo napako za vsak izhodni nevron posebej, zato lahko pozornost enostavno usmerimo z manipulacijo napak posameznih izhodov. Podobno kot neoznačenim delom, ki jim običajno ne dodelimo nič pozornosti in v času učenja zato njihove napake nastavili na vrednost nič, pri obravnavi robov cestišča želimo, da bi slikovni elementi, ki prestavljajo rob cestišča, imeli večjo utež. Njihove napake zato v času učenja množimo z vrednostjo večjo od ena in s tem povečamo vpliv napake teh slikovnih elementov.

Slednje predpostavlja, da poznamo, kje se nahaja rob cestišča. V kolikor imamo natančne oznake, kot so npr. tiste zbrane v zbirki Cityscape, je določanje robov cestišča enostavno. Na področju računalniškega vida za detekcijo robov običajno uporabimo gradient slike, vendar pa nas v našem primeru ne zanima le rob cestišča. Poudarili bi namreč radi rob cestišča, nekoliko manj pa tudi slikovne elemente, ki so v neposredni bližini samega roba. Za določanje obrobnega pasu, ki predstavlja rob cestišča in pločnika (oziroma ozadja), bomo zato uporabili morfološko širitev in erozijo. Obrobni pas tako določimo na sledeč način:

$$\text{obrobni\_pas} = (\text{cesta} \oplus \mathit{strukturni\_el}) - (\mathit{cesta} \ominus \mathit{strukturni\_el}), \quad (3.2)$$

kjer simbol $\oplus$ predstavlja širitev, simbol $\ominus$ erozijo in $-$ razliko med binarnima slikama. Širino obrobnega pasu določamo z obliko in velikostjo strukturnega elementa. V naši rešitvi smo uporabili kvadratni strukturni element velikost 100×100 slikovnih elementov. Izbrali smo ga eksperimentalno na podlagi

lastne presoje. Izbira bi lahko bila kompleksnejša, v kolikor bi temeljila na statistiki povprečne oddaljenosti napačno klasificiranih slikovnih elementov predhodnih rešitev, vendar pa smo se za potrebe demonstracije odločili za enostavnejši pristop.

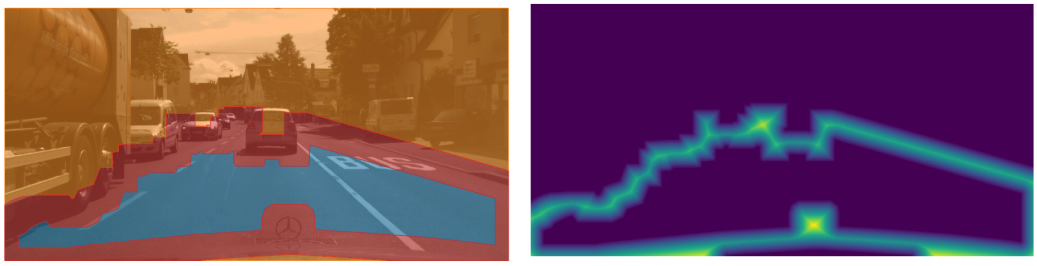
Obrobni pas bi torej želeli primerno utežiti. Pri tem bi radi upoštevali, da so elementi na robovih pasu manj uteženi, kot tisti proti sredini. Za določanje uteži si lahko pomagamo s t.i. transformacijo razdalje (angl. distance transform) [30], ki za vsak slikovni element znotraj obrobnega pasu izračuna evklidsko razdaljo do najbližjega roba cestišča. Slikovne elemente znotraj obrobnega pasu nato utežimo z večkratnikom obratne vrednosti evklidske razdalje, kot je to vidno na Sliki 3.4. Obratno vrednost potrebujemo zato, ker želimo, da imajo slikovni elementi, ki se nahajajo tik ob robu, največjo utež, tisti najdlje pa najnižjo. Novo funkcijo izgube bi tako lahko zapisali na sledeč način:

$$C(\mathbf{w}) = -\sum_{i=1}^{N}\sum_{k=1}^{K} y_{true}^{(k)} * \log(y_{predicted}^{(k)}) * \alpha u^{(k)}, \tag{3.3}$$

kjer $u^{(k)}$ predstavlja utež k-tega slikovnega elementa, α pa večkratnik, ki nam omogoča reguliranje moči uteži. Gre za nov hiper-parameter, ki neposredno vpliva na samo karakteristiko funkcije izgube. Izbira tega parametra je primerljiva z nastavljanjem stopnje regularizacije, zato ga je potrebno določiti eksperimentalno s pomočjo validacijske množice.

Obravnava pomanjkljivih oznak

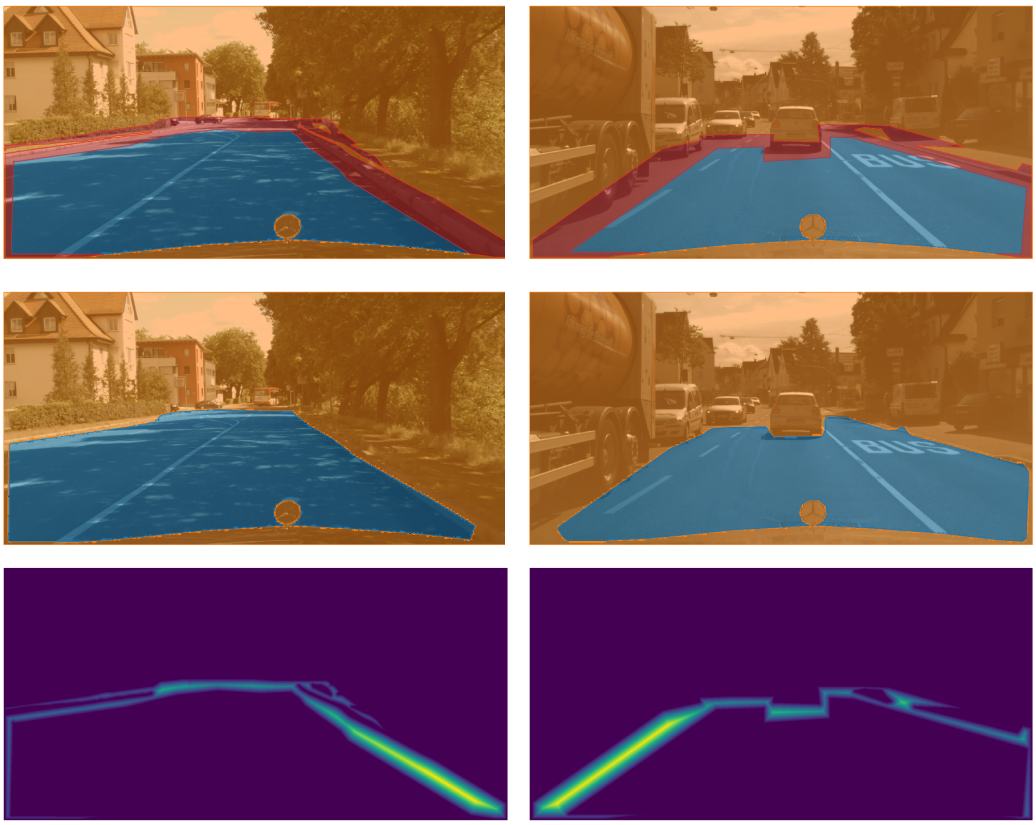
V praksi pogosto nimamo dostopa do natančnih oznak, saj je ročno označevanje segmentacijskih mask dolgotrajna naloga. Zato si označevalci pogosto pomagajo z dodatno kategorijo za neoznačene dele, kot si bomo to podrobneje pogledali v poglavju 4.1.2. Pogosto se za pridobivanje oznak uporabljajo tudi enostavnejši algoritmi računalniškega vida. Ti so običajno manj natančni, zato napovedi slikovnih elementov, za katere algoritem ni izrazil visoke zanesljivosti, ne vključimo v oznako.



Slika 3.4: Leva slika prikazuje obrobni pas obarvan z rdečo barvo, ozadje z oranžno in cesto z modro. Na desni sliki pa vidimo primer uteži slikovnih elementov, ki pripadajo obrobnemu pasu. Tu tople barve (rumena) predstavljajo višjo utež, medtem ko hladne barve (vijolična) predstavljajo nižjo utež [2].

Rezultate eksperimentov si bomo podrobneje pogledali v prihodnjem poglavju, za zdaj pa le omenimo, da je uspešnost nevronske mreže, naučene na grobo označenih podatkih, nekoliko slabša od tiste, naučene na natančno označenih podatkih. Iz tega sledi, da so neoznačeni deli pomembni za nevronsko mrežo. Posledično se lahko vprašamo, ali lahko na kakšen način uporabimo neoznačene dele sliki in razliko med uspešnostjo teh dveh modelov zmanjšamo.

Pristop, ki smo ga obravnavali v naši raziskavi, temelji na uporabi neoznačenih delov, katerim v času učenja dodelimo nižjo utež od označenih. Z vidika uteževanja slikovnih elementov oziroma usmerjanja pozornosti je pristop zelo podoben tistemu, ki smo ga opisali v prejšnjem poglavju. Glavna razlika je, da nam tokrat pripadajoče kategorije neoznačenih slikovnih elementov niso znane. Naš eksperiment temelji na predpostavki, da sosednji slikovni elementi pogosto pripadajo isti kategoriji. Upoštevanje te predpostavke nam omogoča enostaven mehanizem za določanja kategorij neoznačenih predelov slik. Ponovno si lahko pomagamo z transformacijo razdalje, ki nam za vsak slikovni element vrne dve evklidski razdalji. Prva razdalja predstavlja oddaljenost danega slikovnega elementa do najbližjega slikovnega elementa, označenega kot cestišče, druga pa do najbližjega elementa, označenega kot



Slika 3.5: Prikaz dveh učnih primerov (vsak v svojem stolpcu). V prvi vrsti so prikazane pomanjkljive oznake, v drugi vrsti oznake pridobljene z opisanim postopkom in v tretji vrsti uteži novo označenih delov.

ozadje. Neoznačenim slikovnim elementom tako dodelimo kategorijo, ki jo ima najbližji označen slikovni element.

Slika 3.5 prikazuje rezultat opisanega postopka dodeljevanja kategorij neoznačenim delom dveh učnih primerov (vsak v svojem stolpcu). V prvi vrsti se nahaja prikaz, kjer je vhodna slika prekrita z originalno oznako. Oranžna barva predstavlja ozadje, modra cestišče in rdeča neoznačen del slike. V drugi vrsti je prikazana oznaka, pridobljena s pomočjo transformacije razdalje, kjer oranžna barva predstavlja ozadje, modra pa cesto. Zadnja vrsta prikazuje uteži neoznačenih delov, kjer toplejše barve (rumena) v primerjavi s hladnejšimi (vijolična) predstavljajo nižjo utež.

3.3.2 Klasifikacija

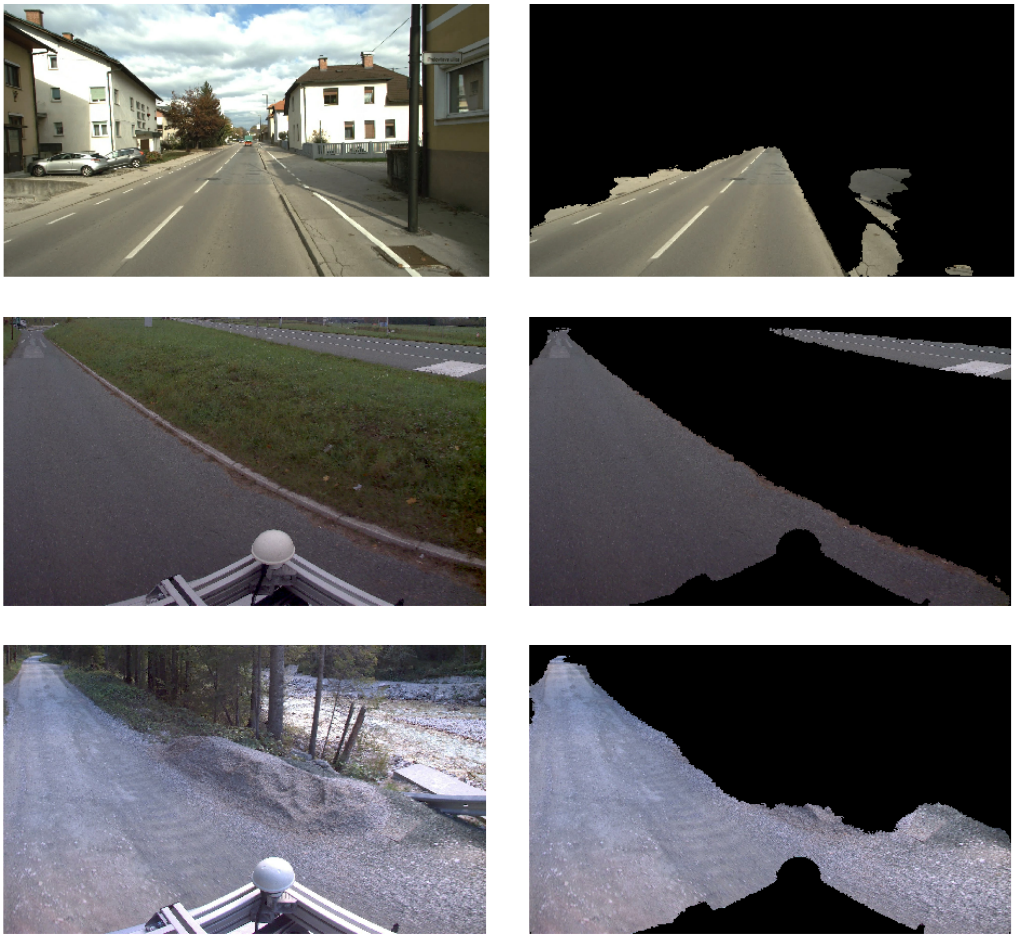
V primerjavi s segmentacijskimi mrežami, ki smo jih spoznali v prejšnjem poglavju, je pristop k usmerjanju pozornosti klasifikacijske mreže nekoliko drugačen. Ker izhod klasifikacijske mreže nima istih dimenzij kot vhodna slika in izhodni nevroni ne odražajo kategorij istoležnih slikovnih elementov, pozornosti ne moramo usmerjati z manipulacijo posameznih komponent funkcije izgube. Za usmerjanje pozornosti si bomo zato pogledali dva različna pristopa, obema pa je skupno, da temeljita na informaciji o položaju cestišča, pridobljeni s pomočjo segmentacije, kot je bilo to predstavljeno v zadnjem poglavju.

Maskiranje z ničelno vrednostjo

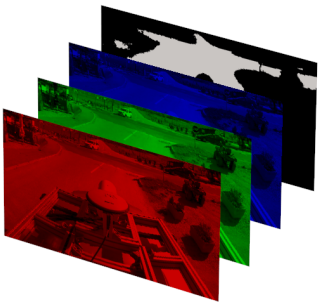
Najpreprostejša zamisel za usmerjanje pozornosti je maskiranje vhoda z ničelno vrednostjo. V tem primeru slikovne elemente vhodne slike, katere je segmentacijska nevronska mreža označila za ozadje, še pred vstopom v nevronsko mrežo nastavimo na nič. Maskiranje z ničelno vrednostjo si lahko predstavljamo, kot zamenjavo ozadja s črno barvo. Vhodna slika tako ne vsebuje informacije, za katero segmentacijska nevronska mreža meni, da ni relevantna. Klasifikacijska nevronska mreža in postopek njenega učenja se v tem primeru ne razlikuje od tistega, ki bi ga uporabili, v kolikor vhodne slike predhodno ne bi maskirali z ničelno vrednostjo. Primer vhodnih slik je možno videti v desnem stolpcu Slike 3.6.

Razširitev z novim kanalom

Z maskiranjem torej izključimo vse, kar ni cestišče. To pa ni vedno najboljša rešitev, saj kontekst lahko nosi veliko relevantnih informacij. V naši domeni je dober primer okolica - makadamske ceste se pogosteje nahajajo v naravi kot pa mestnih središčih. Tako nam kontekst lahko zelo pomaga pri napovedi tipa cestišča in bi ga bilo nesmiselno izločiti. Maskiranje z ničelno vrednostjo zato lahko nadomestimo z razširitvijo vhodne slike z novim kanalom, ki ga



Slika 3.6: Slika prikazuje tri primere vhodnih slik (vsak v svoji vrsti), kjer levi stolpec predstavlja originalno vhodno sliko, desni pa vhodno sliko, maskirano z ničelno vrednostjo.



Slika 3.7: Prikaz združevanja kanalov slike in segmentacijske maske.

predstavlja izhod segmentacijska mreže. Na ta način mreži zagotovimo vso informacijo, ki bi jo sicer dobila z vhodno sliko, hkrati pa jo obogatimo z informacijo o položaju cestišča.

Uporaba tega pristopa ima za razliko od maskiranja z ničelno vrednostjo nekaj posledic na samo arhitekturo nevronske mreže. V tem primeru je vhod drugačnih dimenzij in posledično tudi prvi konvolucijski nivo. Prenos znanja zato ni več trivialen, saj se arhitektura mreže ResNet-50 nekoliko spremeni. Prenos znanja sicer še vedno lahko uporabimo za učne parametre, ki so skupni obema arhitekturama, na novo dodane uteži pa inicializiramo z naključnimi vrednostmi.

Vhoda posledično tudi ni mogoče prikazati, lahko pa si ga enostavno zamislimo kot združitev posameznih kanalov slike in segmentacijske maske, kot je to prikazanih na Sliki 3.7.

Poglavje 4

Zasnova eksperimentov

4.1 Podatkovne zbirke

Kot smo omenili na začetku poglavja 2 potrebujemo za proces iskanja ustreznih parametrov nevronske mreže podatkovno zbirko, ki jo sestavljajo pari vhodnih in izhodnih podatkov. Podatke, ki jih uporabljamo za učenje, imenujemo učni podatki. To pa niso edini podatki, ki jih potrebujemo za reševanje problemov, kot sta klasifikacija in segmentacija. Da se lahko prepričamo, ali je pridobljena rešitev po končanem učenju zares uporabna, moramo samo rešitev evalvirati. Za ocenjevanje uspešnosti potrebujemo pare vhodnih in izhodnih podatkov, katerih nismo uporabili za potrebe učenja, saj bi bila ocena v tem primeru pristranska. Za evalvacijo zato potrebujemo tako imenovane testne podatke. Ker učenje modelov pogosto zahteva eksperimentalno nastavljanje različnih hiper-parametrov (tj.parametrov, ki definirajo lastnosti modela ali spreminjajo proces učenja), v praksi pogosto potrebujemo še podatke za validacijo. Gre za množico podatkov, s pomočjo katere ocenimo primernost izbranih hiper-parametrov. Da je ocenjevanje resnično nepristransko, mora validacijska množica vsebovati podatke, ki niso vsebovani ne v učni ne v testni množici.

Podatkovno zbirko torej lahko razdelimo na tri dele (učno, validacijsko in testno množico), vsak vsebovan primer pa je pravzaprav par vhodnega

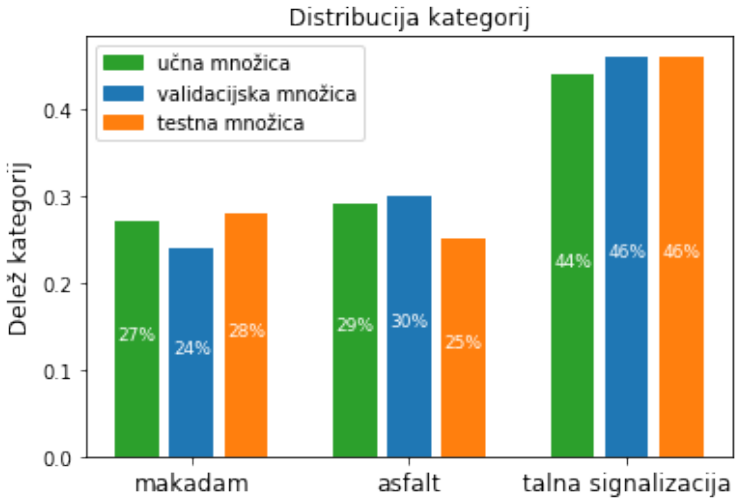
podatka in temeljne resnice. Tako za klasifikacijo kot tudi segmentacijo bo vhodni podatek predstavljala slika, temeljna resnica pa je za klasifikacijo nekoliko drugačen kot za segmentacijo. V nadaljevanju si zato poglejmo, kakšne podatke bomo uporabili za reševanje obeh podproblemov.

4.1.1 Podatki za klasifikacijo

Za naš klasifikacijski problem je izhodni podatek kategorija oziroma razred (angl. class), ki predstavlja tip površine cestišča. Za potrebe učenja moramo kategorije predstaviti na ustrezen način, tako da bo z njimi matematično lažje operirati. V praksi se najpogosteje pojavljata dve možnosti predstavitve: predstavitev s celoštevilskim zapisom (angl. sparse encoding) ali pa bitno kodiranje (angl. one-hot encoding), pri katerem vsako kategorijo predstavimo s svojim bitom, nastavljen na vrednosti ena pa je le tisti bit kategorije, kateri pripada slika (oziroma slikovni element v primeru semantične segmentacije). Pri reševanju našega problema bomo bitno kodiranje uporabili tako za predstavitev klasifikacijskih kot tudi segmentacijskih podatkov.

Za potrebe eksperimentiranja smo označili 2500 slik, vsako z eno od treh kategorij, predstavljeno z bitnim zapisom. Od tega jih 2000 predstavlja učno množico, 250 validacijsko in 250 testno množico. Razredno porazdelitev za vsako izmed množic si je mogoče ogledati na Sliki 4.1. Podatkovno zbirko bomo v nadaljevanju imenovali RST (kot tip vozne površine - angl. Road Surface Type).

Nekaj primerov vhodnih slik je prikazanih na Sliki 4.2. Prva vrsta vsebuje sliki makadamskega, druga vrsta asfaltiranega cestišča in tretja vrsta cestišče s talno signalizacijo. Slike v desnem stolpcu prikazujejo primere, katerih tip površine je težavno določiti že za človeško oko. To so pogosto presvetljene ali osenčene slike, slike na katerih so talne signalizacije na neobičajnih mestih ali pa slika vsebuje dve vozišči različnih tipov. V slednjem primeru smo težavo zaobšli na način, da smo sliki dodelili prevladujočo kategorijo, v kolikor se je to zdelo smiselno.



Slika 4.1: Slika prikazuje distribucijo kategorij za vsako od množic. Zelena barva predstavlja učno množico, modra validacijsko in oranžna testno množico.

4.1.2 Podatki za segmentacijo

Tako kot za klasifikacijo je tudi za problem semantične segmentacije vhodni podatek slika. Zaradi same narave problema pa je izhodni podatek nekoliko drugačen. Semantična segmentacija je razumevanje slikovne vsebine na ravni slikovnih elementov, kar pomeni, da napovedujemo kategorije za vsak slikovni element posebej. Za vsak slikovni element imamo torej svojo kategorijo, ki jo, tako kot pri klasifikaciji, predstavimo ali s celim številom ali pa z bitnim kodiranjem. Temeljno resnico nam zato predstavlja slika kategorij, ki ji večkrat rečemo segmentacijska maska. V našem problemu segmentacije cestišča imamo opravka z dvema kategorijama - cesto in ozadjem (vsem, kar ne predstavlja cestišča). Gre torej za binarno klasifikacijo slikovnih elementov, zato bo segmentacijska maska kar binarna slika, oziroma dve komplementarni sliki v primeru bitnega kodiranja.

Ročna priprava segmentacijskih mask je dolgotrajen proces, hkrati pa tudi težka naloga za označevalca, kajti pogosto določeni deli slike tudi za

Slika 4.2: Prikaz šestih primerov zbirke RST, kjer sliki v prvi vrsti vsebujeta makadamsko cestišče, v drugi vrsti asfaltirano in v tretji asfaltirano cestišče s talno signalizacijo.

človeško oko niso prepoznavni. S tem razlogom se obstoječim kategorijam v praksi pogosto doda posebno kategorijo, ki predstavlja neoznačen del slike (angl. void label). Označevalci kategorijo pogosto uporabljajo za lažje in hitrejše ustvarjanje segmentacijskih mask, ki pa so posledično nekoliko manj natančne. Na tem mestu izpostavimo, da napoved, pridobljena z nevronsko mrežo nikoli ne vsebuje te kategorije, zato problem na ravni slikovnega elementa ostaja binaren, kljub temu da ročno pridelana podatkovna zbirka vsebuje tri kategorije.

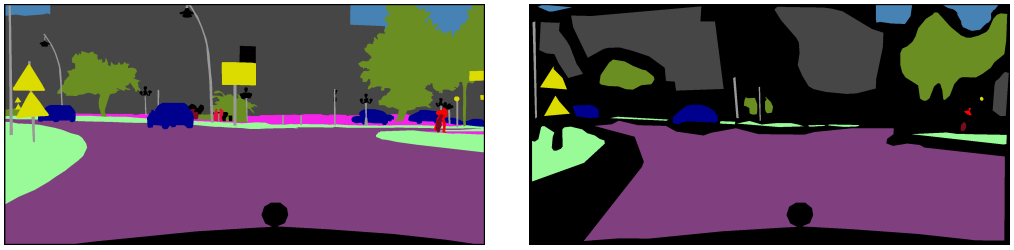
Ker je ročno označevanje segmentacijskih mask časovno zamuden proces, si bomo pomagali s podatkovno zbirko Cityscape [2], ki je javno dostopna za raziskovalne namene. V nadaljevanju si bomo pogledali, kakšne podatke omenjena zbirka vsebuje in kako jih je potrebno predhodno obdelati, da so primernejši za naše eksperimentalne potrebe.

Podatkovna zbirka Cityscape

Podatkovna zbirka Cityscape vsebuje 23.473 različnih slik, ki so razdeljene v tri podmnožice:

- učno množico - vsebuje 2.975 in je namenjena učenju modela
- dodatno učno množico - je prav tako namenjena učenju modela, vsebuje pa 19.998 slik
- validacijsko - vsebuje 500 slik in je namenjena evalvaciji naučenega modela

Za učno in validacijsko množico imamo na razpolago dve vrsti segmentacijskih mask - natančno označene (angl. fine anotations) in grobo označene (angl. coarse anotations). Za dodatno učno množico pa imamo na razpolago le grobo označene segmentacijske maske. V nadaljevanju bomo uporabljali kratico CSF (krajše za Cityscape Fine) za množico, ki vsebuje natančno označene slike učne množice. S kratico CSC (krajše za Cityscape Coarse) bomo označili množico podatkov učne množice z grobo označenimi temeljnimi resnicami. Kratica CSCE (krajše za Cityscape Coarse Extras) pa bo



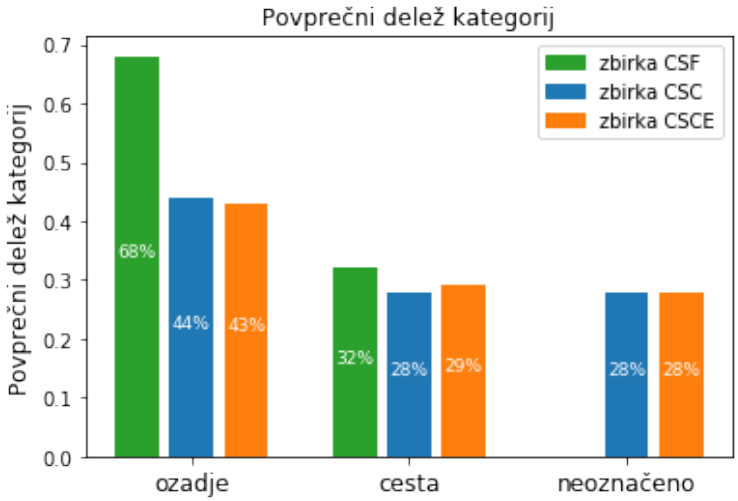
Slika 4.3: Primer natančno označene (levo) in grobo označene segmentacijske maske (desno), ki pripadata isti sliki.

predstavljala množico podatkov iz dodatne učne množice, ki prav tako vsebujejo grobo označene temeljne resnice.

Primer natančno in grobo označenih segmentacijskih mask je prikazan na Sliki 4.3.

Tako natančno označene kot tudi grobo označene segmentacijske maske vsebujejo 30 kategorij. Za nas so pomembne le tri - cesta, ozadje in kategorija, ki predstavlja neoznačene slikovne elemente. Vse označene slikovne elemente, ki niso označeni kot cesta, bomo zato obravnavali kot ozadje. Ko združimo vse irelevantne kategorije z ozadjem, se pojavljajo neoznačeni deli slike tudi na mestih, ki bi jih sicer enostavno razglasili za ozadje, v kolikor bi podatke označevali sami. Pojav je možno opaziti na Sliki 4.6.

Analiza učne in dodatne učne množice pokaže, da je med grobo označenimi segmentacijskimi maskami kar tretjina slikovnih elementov neoznačenih. Delež kategorij je prikazan na Sliki 4.4, kjer zelena barva prikazuje distribucijo kategorij za natančno označene temeljne resnice podatkov učne množice, modra barva distribucijo grobo označene iste učne množice in oranžna barva distribucijo razredov grobo označenih temeljni resnic dodatne učne množice. Statistika je bila izračunana na podlagi 1000 primerov za vsako od treh omenjenih zbirk. Opazimo lahko, da so kategorije na grobo označenih maskah približno enakomerno zastopane, pri natančno označenih podatkih pa je delež elementov, ki predstavljajo ozadje, bistveno večji. Sklepati je možno, da je delež ozadja večji na račun manjšega deleža neoznačenih slikovnih elementov, oziroma da večino neoznačenih slikovnih elementov v resnici predstavlja



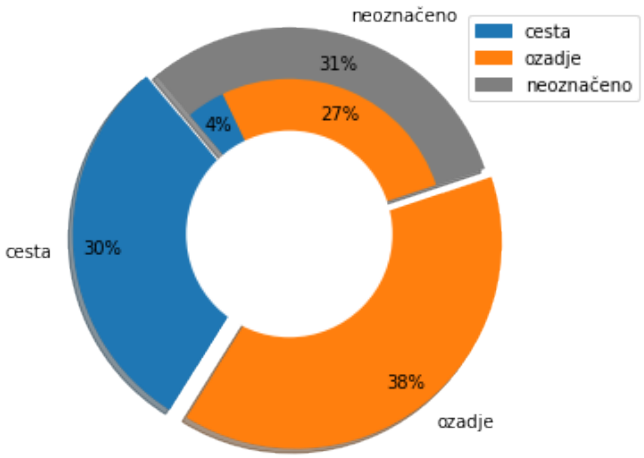
Slika 4.4: Prikaz razredne porazdelitve za vsako od treh omenjenih množic.

ozadje. Da pa je temu res tako, uporabimo učno in validacijsko množico, ki vsebujeta tako natančno kot tudi grobo označene maske ter opravimo analizo neoznačenih podatkov.

Ker imamo za učno množico zbirke Cityscape na razpolago tako natančno označene kot tudi grobo označene temeljne resnice, lahko naredimo analizo resničnih kategorij neoznačenih delov grobih temeljnih resnic. Slika 4.5 prikazuje primerjavo deleža kategorij glede na tip označevanja. Iz slike je možno razbrati, da dodatnih 27% slikovnih elementov predstavlja cesto, kar je približno 86% vseh neoznačenih podatkov. Če deleža seštejemo, vidimo, da v povprečju približno 65% slike predstavlja ozadje, kar se sovpada z rezultati prikazanimi na Sliki 4.4

Da bi porazdelitev grobo označenih mask približati distribuciji natančno označenih, si lahko pomagamo z morfološkimi operacijami in tako zmanjšamo neoznačene luknje, ki so nastale ob združevanju odvečnih kategorij z ozadjem.

Naša želja je spremeniti le določene neoznačene dele v ozadje, zato bomo masko ceste pustili nedotaknjeno. Prav tako bomo ohranili neoznačene dele, ki se nahajajo blizu robov cestišča, saj tam obstaja večja možnost, da pravzaprav predstavljajo cesto. Spremenili bomo torej le tiste neoznačene dele,

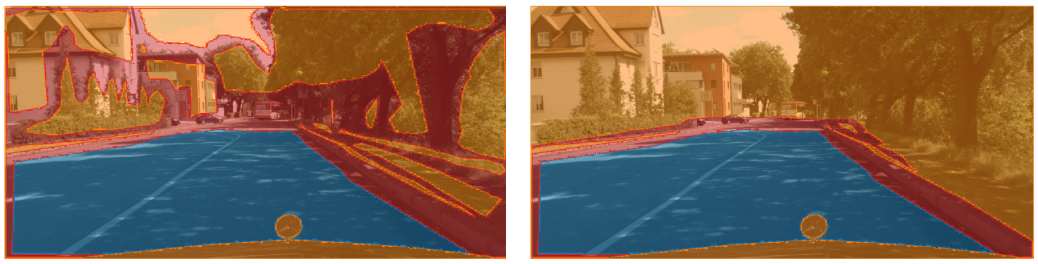


Slika 4.5: Slika prikazuje primerjavo razrednih deležev natančno in grobo označenih temeljnih resnic istih slik. Statistika je bila izračunana na podlagi 1000 primerov.

ki niso blizu ceste. To dosežemo tako, da v prvem koraku na maski ceste uporabimo morfološko širitev (angl. morphological dilation) ter s tem pridobimo kritične dele slike, kjer je večja možnost, da so neoznačeni elementi v resnici cesta. Nato poiščemo presek med dobljeno masko in obstoječo masko neoznačenega dela slike. Elemente, ki niso v preseku in pripadajo prvotni maski neoznačenega dela, razglasimo za ozadje, medtem ko tiste v preseku obravnavamo kot novo masko neoznačenih slikovnih elementov, ki jo bomo v nadaljevanju imenovali erodirana segmentacijska maska neoznačenih slikovnih elementov. Erodirana zato, ker rezultat spominja na morfološko erozijo (angl. morphological erosion) maske neoznačenega dela slike, vendar pa je potrebno poudariti, da bi z uporabo morfološka erozija dobili nekoliko drugačen rezultat.

Slika 4.6 prikazuje primerjavo med prvotno segmentacijsko masko in masko pridobljeno po predstavljenem postopku.

Vse grobo označene maske iz učne in validacijske množice lahko obdelamo po omenjenem postopku in novo statistiko primerjamo s predhodno. Novo



Slika 4.6: Primer originalne (levo) in erodirane segmentacijske maske (desno).

statistiko lahko vidimo na Sliki 4.7.

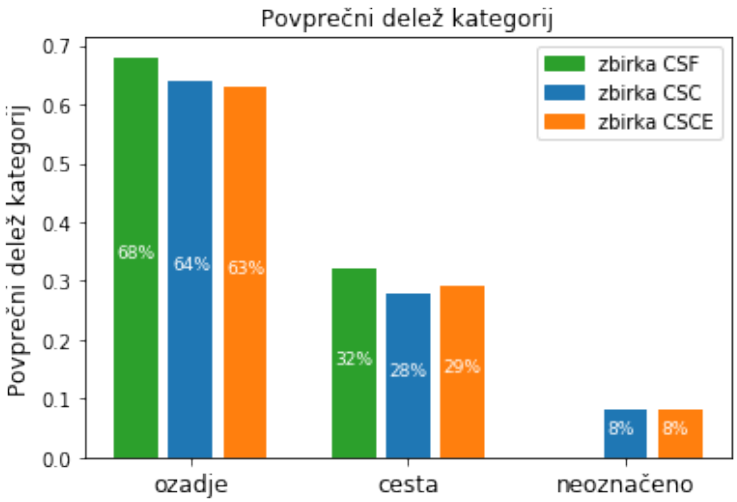
Prav tako lahko izračunamo napako, ki smo jo naredili pri zgoraj opisanem postopku. Od vseh novo označenih primerov nam napako predstavlja delež tistih, ki so bili napačno klasificirani. To so tisti primeri, ki so bili na grobo označeni maski neoznačeni, na natančno označeni so pripadali cesti, mi pa smo jih določili za ozadje. Napaka, izračunana na podlagi tisočih slik, za katere imamo na razpolago tako grobe kot tudi natančne oznake, znaša 1.13% vseh slikovnih elementov, ki smo jim tekom postopka spremenili kategorijo.

V poglavju 3.3.1 smo opisali tehniko obravnavanja pomanjkljivih oznak s pomočjo transformacije razdalje, ki jo bomo večkrat uporabili v eksperimentih. Slika 4.8 prikazuje postopek dodeljevanja kategorij neoznačenim delom. Opazimo lahko, da pridobljena oznaka pogosto vsebuje napake. Ker gre za raziskavo in imamo na razpolago tudi natančne oznake, imamo možnost, da lahko opravimo primerjavo pridobljenih oznak z natančnimi in analiziramo tipe storjenih napak. Te možnosti v praksi sicer pogosto ne bi imeli.

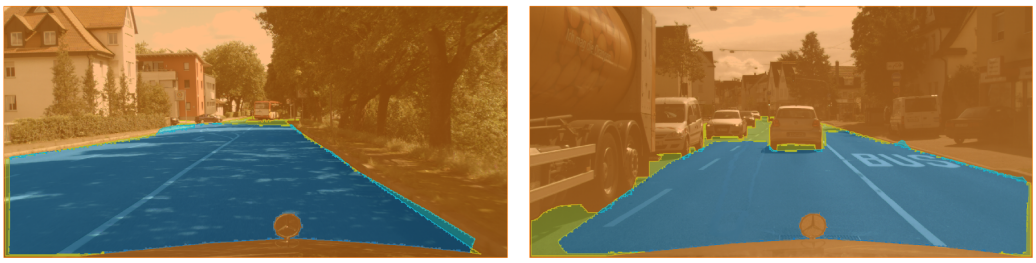
Relativne napake so prikazane s pomočjo matrike zamenjav na Sliki 4.9, kjer lahko opazimo, da smo v 22% ozadje nepravilno označili za cesto in v 31% cesto nepravilno označili za ozadje

Podatkovna zbirka RSTS

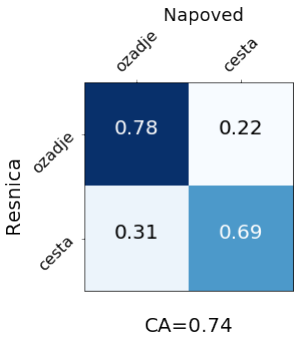
Ker so slike zbrane v podatkovni zbirki Cityscape precej drugačne od tistih v zbirki RST, ki jih bomo uporabili za klasifikacijo, smo za potrebe segmen-



Slika 4.7: Ponoven prikaz razrednih porazdelitev za vsako od množic, kjer tokrat zbirki CSC in CSCE vsebujeta erodirane maske.



Slika 4.8: Prikaz pridobljene napake pri označevanju neoznačenih delov slike s pomočjo transformacijo razdalje.



Slika 4.9: Prikaz matrike zamenjav pridobljene na podlagi 1000 primerov, kjer se za izračun uporabi le neoznačene slikovne elemente (slikovne elemente, katerim smo z transformacijo razdalje dodelili kategorijo).

tacije ročno označili dodatnih 600 slik, pridobljenih z isto opremo kot tiste v zbirki RST. Od tega 70 slik predstavlja validacijsko množico, 70 testno množico in 460 učno množico. V poglavju 3.1.2 smo si že pogledali, kako lahko znanje pridobljeno z učenjem nevronske mreže na eni zbirki prenesemo in doučimo nevronsko mrežo z uporabo druge učne množice. To zbirko bomo v nadaljevanju imenovali RSTS (krajše za Road Surface Type for Segmentation). Sliki 4.11 in 4.10 prikazujeta nekaj primerov slik zbirke RSTS.

V praksi učenje z več podatki pogosto prinese boljšo uspešnost, po drugi strani pa se s tem čas učenja poveča. Ker je učenje nevronskih mrež samo po sebi dolgotrajen proces, hkrati pa ima veliko nastavljivih hiper-parametrov, kar pomeni večkratno učenje, bomo za potrebe eksperimentiranja uporabili le podmnožico zbirke Cityscape. Hkrati je uporaba manjše učne množice tudi nekoliko bolj realna slika problemov, s katerimi se v praksi pogosto srečujemo. Tako smo za potrebe eksperimentiranja uporabili le del omenjenih zbirk. Iz zbirk CSF in CSC smo vzeli istih 1200 primerov, ki pa se razlikujejo le v tipu oznak (CSF vsebuje natančne oznake, CSC pa grobe). Zbirko CSCE smo sestavili tako, da smo zbirki CSC dodali še dodatnih 1000 slik z grobimi oznakami. Vsi rezultati prikazani v poglavju 4 so povprečja evalvacij petih modelov, katere smo naučili na petih različnih razdelitvah 1200 učnih

Slika 4.10: Prikaz nekaj primerov slik zbirke RSTS.

Slika 4.11: Prikaz dveh primerov zbirke RSTS, kjer je robove cestišča nekoliko težje določiti.

Zbirka	Učna množica		Validacijska množica		Testna množica	
	Oznake	**Št. slik**	**Oznake**	**Št. slik**	**Oznake**	**Št. slik**
CSF	Natančne	1000	Natančne	200	Natančne	400
CSC	Grobe	1000	Grobe	200	Natančne	400
CSCE	Grobe	2000	Grobe	200	Natančne	400
RSTS	Grobe	459	Grobe	70	Grobe	70

Tabela 4.1: Povzetek zbirk in njihovih podmnožic.

primerov oz. 2200 v primeru zbirke CSCE. Od teh je bilo 200 naključno izbranih primerov namenjenih validaciji, preostali pa so predstavljali učno množico. Vse modele smo evalvirali na testni množici, katero smo sestavili iz 400 slik z natančnimi oznakami in niso bile vsebovane v nobeni drugi zbirki. Za lažjo primerjavo rezultatov je testna množica vedno vsebovala iste slike z natančnimi oznakami, razen v primeru zbirke RSTS, kjer smo vse slike ročno označili z grobimi oznakami. Povzetek zbirk oziroma razdelitev na podmnožice si je možno ogledati v Tabeli 4.1.

4.1.3 Umetno bogatenje učne množice

Kot smo že večkrat omenili, je za učenje kompleksnih modelov kot so konvolucijske nevronske mreže predpogoj dostopnost ogromne količine raznolikih podatkov. Popularne zbirke kot je Cityscape pogosto vsebujejo po več tisoč ali deset tisoč učnih primerov. Pogosto nam toliko podatkov ni na voljo, ali pa je proces označevanja predrag, prezahteven ali časovno preveč potraten, zato se v praksi pogosto uporablja tehnika umetnega bogatenja podatkovne zbirke (angl. data augmentation)[31]. Gre za manipulacijo slik in oznak obstoječe zbirke s pomočjo enostavnih operacij, tako da rezultat predstavlja nov učni primer, ki se razlikuje od ostalih učnih primerov. Enostavne operacije, ki se najpogosteje uporabljajo so zrcaljenje, povečava, translacija, spremembe osvetljenosti, ipd.

Na tem mestu izpostavimo, da je poleg manipulacije vhodne slike potrebno ustrezno prilagoditi tudi pripadajočo oznako. V primeru klasifikacijskih oznak ta korak ni potreben, saj se tip cestišča z uporabo omenjenih preprostih operacij ne spremeni. Nekoliko več pozornosti je potrebno posvetiti segmentacijskim oznakam, kjer je kategorija definirana za vsak slikovni element posebej. Nekatere operacije (npr. rotacija ali zrcaljenje) neposredno vplivajo na razred manipuliranih slikovnih elementov, medtem ko uporaba določenih operacij (npr. spreminjanje osvetljenosti) ne potrebuje dodatnega prilagajanja segmentacijske maske.

Umetni nevronski mreži sintetično modificiran primer predstavlja drugačen učni primer. S tehniko bogatenja zbirke tako omogočimo, da nevronska mreža med učenjem vidi primere, ki sicer niso bili prisotni v originalni podatkovni zbirki. Na ta način postane nevronska mreža bolj robustna oziroma invariantna na spremembe, kot so translacija, rotacija, velikost, ipd. Uporaba omenjenih operacij je v praksi priljubljena predvsem zaradi preproste implementacije. Hkrati pa je uporaba samo enostavnih operacij za umetno bogatenje zbirke nekoliko omejena, saj z njimi lahko generiramo nove primere, ki so relativno podobni obstoječim. Pogosto pa bi zbirko radi obogatili tudi s primeri, ki se bistveno razlikujejo od vseh originalnih učnih primerov. Za-

Slika 4.12: Na levi strani je prikazana originalna slika, desno pa slika pridobljena s pomočjo nevronskega prenosa stila [33].

mislimo si, da podatkovna zbirka vsebuje le slike cestišča, ki so bile posnete sredi sončnega dne. V zbirki tako primanjkuje primerov cestišča v nočnih ali deževnih razmerah. V kolikor taki podatki niso prisotni v času učenja, je smiselno predvidevati, da bo razpoznavanje cestišča na nočnih posnetkih manj uspešno. Posledično je danes možno najti veliko raziskav na temo kompleksnejših metod umetnega bogatenja podatkovnih zbirk. En od pristopov je uporaba tako imenovanega nevronskega prenosa stila (angl. neural style transfer)[32], kjer s pomočjo nevronske mreže prenesemo stil ene domene na sliko iz druge domene. Tako bi lahko slike, posnete poleti, s pomočjo nevronskega prenosa stila transformirali v slike, podobne tistim, ki bi bile posnete v snežnih zimskih razmerah.

S kompleksnejšimi pristopi se v praksi srečujemo redkeje, saj terjajo zahtevno implementacijo prenosa stila, hkrati pa so računsko bolj zahtevni. V kolikor bi razvijali rešitev za avtonomna vozila, bi bil razmislek o uporabi kompleksnejših pristopov bogatitve podatkovne zbirke smiseln. Za rešitev, ki jo potrebujejo vzdrževalci cestišč pa nam zadostuje že uporaba preprostih operacij, saj lahko predpostavljamo, da so slike cestišč zajete tekom delavnega časa ter ob lepem vremenu. V nadaljnjih eksperimentih smo med

učenjem uporabili tehniko umetnega bogatenja učne množice, da pa bi bili rezultati med seboj primerljivi, sam pred vsakim začetkom učenja fiksirali vse parametre generatorjev naključnih števil. Operacije, ki smo jih uporabili za umetno obogatitev učne množice so bile:

- horizontalno zrcaljenje v 50% primerih,
- rotacija okoli središča slike za naključni kot do pet stopinj v obratni smeri ali smeri urinega kazalca,
- povečava slike za naključni faktor med 0,9 in 1,1,
- premik slike za do 10% v vertikalni in za do 10% v horizontalni smeri.

4.2 Ocenjevanje uspešnosti

Kot smo že omenili v začetku poglavju, uspešnost vedno ocenjujemo s pomočjo testne množice (angl. test set), katero sestavljajo primeri, ki jih nevronska mreža med učenjem ne vidi. V tem poglavju si bomo pogledali nekaj preprostih metrik, ki se pogosto uporabljajo na področju klasifikacije. Za lažjo razlago si bomo najprej ogledali izračun metrik na primeru segmentiranja cestišča, kjer je problem binaren. Nato si bomo pogledali še, kako lahko iste metrike uporabimo tudi za evalvacijo rešitev nebinarnih problemov, kot je klasifikacija tipa cestišča.

4.2.1 Uspešnost segmentacije

Ker segmentacija predstavlja klasificiranje slikovnih elementov, je na posamezni sliki pravzaprav *višina*$\times$*širina* primerov, ki jih je potrebno klasificirati v eno izmed dveh skupin - cesto ali ozadje. Gre torej za binarno klasifikacijo, zato si pri predstaviti uspešnosti lahko pomagamo z matriko zamenjav (angl. confusion matrix). Če izberemo, da cesta predstavlja pozitivni in ozadje negativni razred, lahko definiramo resnično pozitivni primer (angl. true positive) kot slikovni element, ki na sliki predstavlja cesto, kot cesto

		Napoved	
		Cesta	**Ozadje**
Resnica	**Cesta**	resnično pozitivni	lažno negativni
	Ozadje	lažno pozitivni	resnično negativni

Tabela 4.2: Prikaz matrike zamenjav.

pa ga je prav tako klasificirala nevronska mreža. Lažno negativni primer (angl. false negative) prav tako predstavlja primer, kjer slikovni element pripada cesti, vendar pa ga je nevronska mreža napačno klasificirala za ozadje. Na podoben način definiramo še resnično negativni (angl. true negative) in lažno pozitivni (angl. false positive) primer, kjer je nevronska mreža pravilno oziroma napačno dodelila kategorijo slikovnemu elementu, ki na sliki predstavlja ozadje. Matriko zamenjav tako sestavimo iz teh štirih statistik, kjer za izračun vsake statistike uporabimo vseh *št. slik* × *višina* × *širina* primerov. Matrika zamenjav je prikazana s tabelo 4.2.

Iz omenjenih štirih statistik lahko izpeljemo veliko uporabnih metrik, a tu bomo omenili le štiri, katere bomo uporabljali za primerjavo različnih naučenih modelov.

Klasifikacijska točnost (angl. Classification Accuracy) je verjetno najbolj prepoznavna matrika. Definirana je kot delež pravilno klasificiranih primerov, oziroma z enačbo:

$$CA = \frac{TP + TN}{TP + TN + FP + FN}. \tag{4.1}$$

Priklic ali senzitivnost (angl. recall ali sensitivity) je delež pozitivnih primerov, ki so bilo pravilno klasificirani kot pozitivni. V primeru segmentacije ceste priklic predstavlja delež ceste, ki smo ga pravilno označili z oznako

cesta. Z enačbo priklic zapišemo kot:

$$priklic = \frac{TP}{TP + FN}. \tag{4.2}$$

Preciznost ali pozitivna prediktivna vrednost (angl. precision ali positive predictive value - krajše PPV) se izračuna na podoben način. Gre za razmerje med pravilno klasificiranimi pozitivnimi primeri in vsemi primeri, ki so bili klasificirani kot pozitivni. Z enačbo to zapišemo na sledeč način:

$$PPV = \frac{TP}{TP + FP}. \tag{4.3}$$

S preciznostjo in priklicem tako lahko opišemo uspešnost poljubnega binarnega klasifikatorja. Pogosto pa radi primerjamo uspešnost različnih naučenih modelov, kar pomeni, da je potrebno primerjati preciznosti in priklic vsakega modela. Da bi bila primerjava enostavnejša definiramo eno samo mero - tako imenovano mero F1 (angl. F1 score). Ta predstavlja harmonično povprečje priklica in preciznosti, matematično pa jo zapišemo s formulo:

$$F1 = \frac{2 \times preciznost \times priklic}{preciznost + priklic}. \tag{4.4}$$

Zato da je ocena uspešnosti za poljuben modela realistična, mora testna množica vsebovati veliko slik. Ker pri segmentaciji za vsako vhodno sliko klasificiramo vse njene slikovne elemente, najprej izračunamo opisane metrike za vsako vhodno sliko posebej, nato pa rezultate povprečimo s številom slik. Pridobljen rezultat nam tako predstavlja oceno povprečne uspešnosti našega modela.

Na tem mestu omenimo še obravnavno neoznačenih slikovnih elementov. Kadar slikovni element nima pripadajoče oznake, za napoved ne moramo trditi, da je pravilna ali napačna, zato za izračun omenjenih metrik neoznačenih slikovnih elementov običajno ne obravnavamo.

Ker segmentacijski problem na eni sami sliki vsebuje toliko učnih primerov, kot je ločljivost slike, je matriko zamenjav, pridobljeno na evalvaciji ene

Slika 4.13: Primer vizualizacije metrike razporeditev pridobljeno z evalvacijo ene same vhodne slike.

same slike, možno lepo vizualizirati. S tako vizualizacijo se bomo srečevali skozi celo nalogo, zato si poglejmo, kako vizualizacija izgleda in kako jo razumeti. Slika 4.13 prikazuje vizualizacijo metrike razporeditev pridobljeno z evalvacijo ene same vhodne slike. S temno modro so obarvani pravilno klasificirani pozitivni primeri (torej pravilno zaznana cesta). Z oranžno barvo so obarvani pozitivno negativni primeri (pravilno zaznano ozadje). Svetlo modra barva predstavlja lažno pozitivne primere (ozadje, ki smo ga napačno označili za cesto). Z rumeno pa so obarvani lažno negativni primeri (napačno označena cesta). Poleg štirih barv občasno zasledimo tudi sivo barvo, s katero opišemo neoznačene dele slike.

4.2.2 Uspešnost klasifikacije

Za ocenjevanje uspešnosti napovedovanja tipa cestišča bomo uporabljali metrike, ki smo jih pravkar predstavili. Izračun pa se bo v tem primeru nekoliko razlikoval, saj se problem klasifikacije tipa cestišča od segmentacije ceste razlikuje v dveh aspektih.

Prva razlika je v številu primerov glede na posamezno vhodno sliko. Pri segmentaciji klasificiramo toliko primerov, kolikor je ločljivost vhodne slike, medtem ko pri napovedovanju tipa cestišča nam vhodna slika predstavlja en sam primer. Izračun metrik se zato nekoliko poenostavi, saj metrik ne računamo za vsako sliko posebej, temveč kar za celotno testno množico.

Druga ključna razlika je v številu razredov, ki jih napovedujemo pri klasifikaciji tipa cestišča. Medtem ko je problem segmentacije binarni problem, bomo tip cestišča uvrščali v enega od treh kategorij. Ker problem ni binaren, se tudi matrika zamenjav nekoliko razlikuje od tabele 4.2 prikazane v prejšnjem poglavju. Kadar imamo več kot dva razreda, lahko problem za potrebe evalvacije binariziramo. V našem primeru to pomeni, da izberemo prvi razred, ga proglasimo za pozitivnega, ostala razreda pa obravnavamo kot negativni razred. Nato izračunamo omenjene metrike, kot bi to naredili za binarni problem. Postopek nato ponovimo, le da za pozitivni razred izberemo drugi razred in navsezadnje še za tretji razred. Tako za vsako metriko dobimo tri rezultate, ki jih pogosto povprečimo, kar pa nam nato predstavlja oceno povprečne uspešnosti našega klasifikatorja.

Poglavje 5

Eksperimentalni rezultati

V tem poglavju si bomo pogledali, kakšne rezultate dobimo, v kolikor uporabimo metode, opisane v prejšnjih poglavjih.

V prvem delu bomo raziskali, kakšno uspešnost dosežemo z uporabo nevronske mreže U-Net, ki za kodirnik uporablja ResNet-50. Pogledali si bomo tudi rezultate pridobljene z različnim uteževanjem slikovnih elementov in rezultate, ki jih dobimo s pomočjo prenosa znanja, pridobljenega z učenjem mreže na zbirki Cityscape in doučeno na zbirki RSTS.

V drugem delu pa se bomo posvetili klasifikaciji tipa površine cestišča. Pogledal si bomo klasifikacijsko uspešnost mreže ResNet-50. Nato bomo pridobljene rezultate primerjali z rezultati dveh eksperimentov, v katerih smo klasifikacijski mreži pomagali z informacijo o položaju cestišča, ki nam jo je vrnila segmentacijska mreža.

5.1 Segmentacija

5.1.1 Natančne oznake

Cilj prvega eksperimenta je poiskati izhodiščno vrednost uspešnosti, s katero lahko nato primerjamo rezultate vseh nadaljnjih eksperimentov. S tem namenom smo naučili dve nevronski mreži. Obe mreži sta arhitekture U-Net, kjer prva za kodirnik uporablja mrežo ResNet-50, druga pa modificiran

Model	Preciznost	Priklic	F1
U-Net ResNet-50	0.9821	**0.9700**	0.9747
U-Net DRN-ResNet-50	**0.9858**	0.9677	**0.9754**

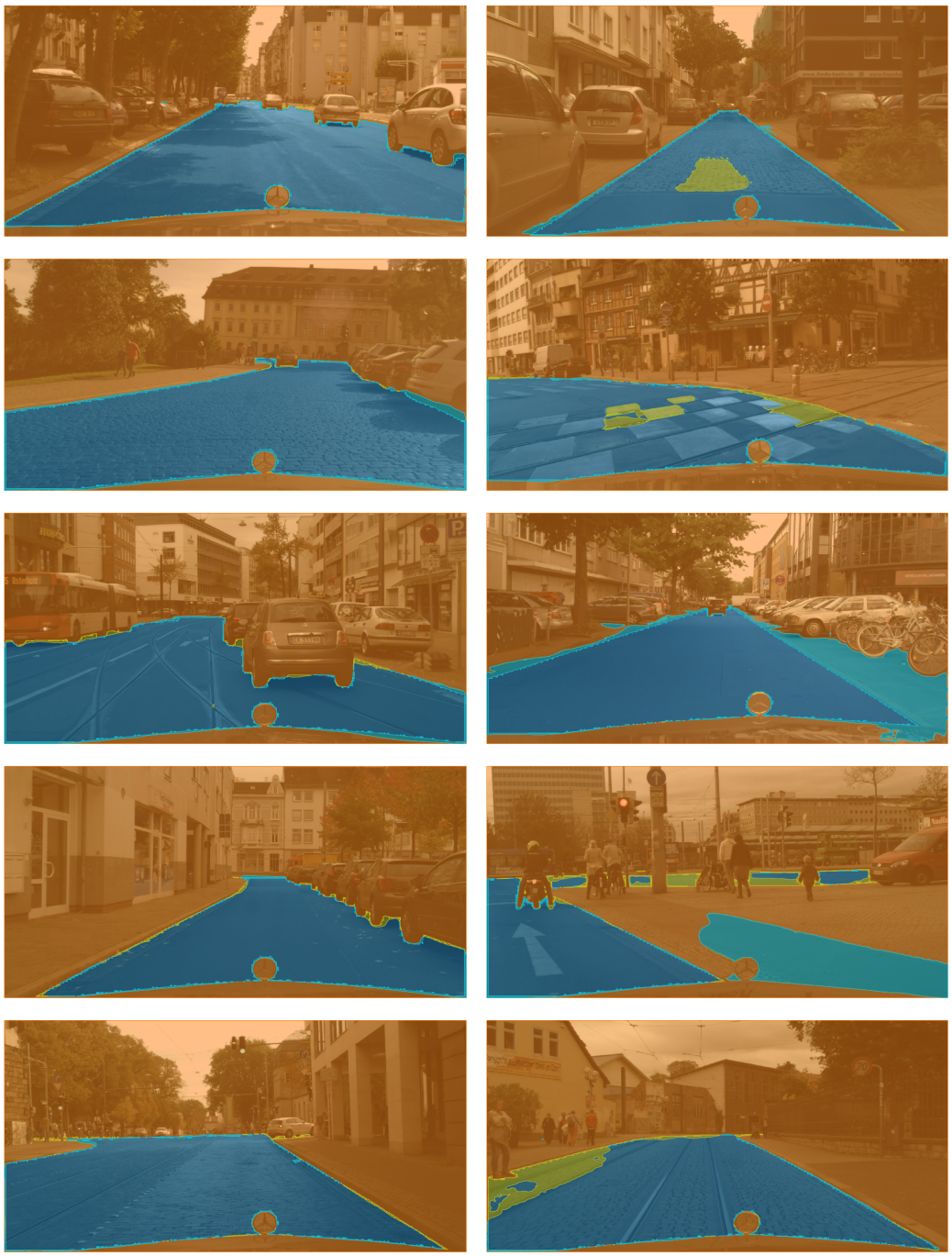
Tabela 5.1: Prikazuj uspešnosti dveh nevronskih mrež arhitekture U-Net, naučenih s pomočjo zbirke CSF (natančno označeni podatki). Prvi model za kodirnik uporablja mrežo ResNet-50, medtem ko drugi model uporablja modificiran ResNet-50.

ResNet-50 z razširjenimi konvolucijami, kot je to opisano v članku [24]. Iz Tabele 5.1 je možno razbrati, da z uporabo razširjene konvolucije dosežemo malenkost boljše rezultate (višja vrednost metrike F1), zato smo v nadaljnjih segmentacijskih eksperimentih za kodirnik uporabili mrežo ResNet-50, ki uporablja razširjeno konvolucijo.

Na Sliki 5.1 je zbranih nekaj primerov segmentacije, pridobljene z mrežo U-Net, ki za kodirnik uporablja ResNet-50 z razširjeno konvolucijo, naučeno na zbirki CSF.

5.1.2 Uteževanje obrobnega pasu

Naslednji eksperiment je predstavljal poizkus izboljšanja uspešnosti z večjim uteževanjem obrobnega pasu. Naučili smo šest nevronskih mrež arhitekture U-Net, kjer smo v času učenja za obrobni pas uporabili različne utežitvene koeficiente, kot smo to opisali v poglavju 3.3.1. V Tabeli 5.2 si je možno ogledati uspešnost vseh šestih nevronskih mrež, naučenih s pomočjo zbirke CSF. Vsaka vrstica tabele prikazuje uspešnost modela naučenega na istih podatkih, kjer smo za poudarek robov cestišča uporabili drugačno utež (prikazano v prvem stolpcu tabele). Iz tabele je možno razbrati, da smo najboljši rezultat pridobili z utežitvenim koeficientom enakim vrednosti šest. V primerjavi z uspešnostjo nevronske mreže, naučene na podatkih, kjer robov cestišča nismo poudarili (prikazano v Tabeli 5.1), lahko vidimo, da smo s poudarkom



Slika 5.1: Slika prikazuje nekaj primerov uspešne (levi stolpec) in neuspešne (desni stolpec) segmentacije na zbirki CSF.

Utežitveni koeficient	**Preciznost**	**Priklic**	**F1**
1	0.9810	0.9665	0.9724
2	0.9844	0.9728	0.9776
3	0.9822	0.9715	0.9754
4	0.9845	0.9737	0.9779
5	0.9823	0.9702	0.9748
6	**0.9854**	**0.9737**	**0.9785**

Tabela 5.2: Prikaz uspešnosti šestih nevronskih mrež arhitekture U-Net, naučenih s pomočjo zbirke CSF z različnimi utežitvenimi koeficienti robov cestišča.

robov dosegli nekoliko boljšo uspešnost, čeprav razlika ni bistvena.

5.1.3 Grobe oznake

V drugem sklopu eksperimentov smo uporabili grobe oznake. Tu smo primerjali, ali uporaba večjega števila učnih primerov izboljša uspešnost naučenega modela. Tabela prikazuje uspešnost dveh nevronskih mrež arhitekture U-Net, kjer smo prvo nevronsko mrežo naučili na zbirki z manj učnimi primeri (zbirka CSC), drugo pa na zbirki CSCE, ki vsebuje več podatkov (kot je to opisano v poglavju 4.1.2). Iz tabele je razvidno, da se uspešnost modela (vrednosti metrike F1) ne razlikujeta bistveno in da z uporabo večje količine grobo označenih podatkov uspešnosti nismo izboljšali.

5.1.4 Utežene grobe oznake

Ker je v povprečju skoraj tretjina grobe oznake neoznačene, nas je zanimalo, ali lahko izboljšamo uspešnost modela, v kolikor upoštevamo tudi neoznačene dele, ki jim dodelimo nižjo utež. S tem namenom smo izvedli osem eksperimentov, kjer smo neoznačenim dodelili oznako s pomočjo transformacijo

Zbirka	Preciznost	Priklic	F1
CSC	**0.9585**	0.9455	0.**9496**
CSCE	0.9486	**0.9578**	0.9491

Tabela 5.3: Prikaz uspešnosti dveh nevronskih mrež arhitekture U-Net, naučenih s pomočjo pomanjkljivo označenih podatkov.

Zbirka	Utežitveni koeficient	Preciznost	Priklic	F1
CSC	1	0.9509	0.9589	0.9525
CSC	0.5	0.9601	0.9495	0.9525
CSC	**0.33**	**0.9599**	**0.9495**	**0.9525**
CSC	0.25	0.9610	0.9460	0.9512
CSC	0.1	0.9496	0.9569	0.9509
CSCE	0.5	0.9585	0.9443	0.9492
CSCE	1	0.9629	0.9407	0.9493
CSCE	**0.33**	**0.9556**	**0.9519**	**0.9515**

Tabela 5.4: Prikaz uspešnosti nevronskih mrež arhitekture U-Net, kjer smo pred učenjem pomanjkljivo označene dele oznake dopolnili z uporabo transformacije razdalje.

razdalje, kot je to opisano v poglavju 3.3.1. Tabela 5.4 prikazuje uspešnosti petih konvolucijskih mrež arhitekture U-Net, naučenih na zbirki CSC in tri mreže iste arhitekture, naučene na zbirki CSCE, kjer smo eksperimentirali z različnimi vrednostmi utežitvenega koeficienta. Iz tabele je možno razbrati, da najboljše rezultate pridobimo, ko imamo utežitveni koeficient nastavljen na vrednost 0,33 (najvišja vrednost F1 za obe zbirki).

V primerjavi s Tabelo 5.3 je možno opaziti, da z uporabo neoznačenih delov slik, ki smo jih označili s pomočjo transformacije razdalje, dosežemo nekoliko boljše rezultate, kot z modeli, ki so v času učenja neoznačene dele

slike ignorirali. Uspešnost sicer ni bistveno večja in se še vedno bistveno razlikuje od uspešnosti modelov, naučenih na natančno označenih podatkih (zbirki CSF), prikazanih v Tabeli 5.1.

5.1.5 Prenos znanja na zbirko RSTS

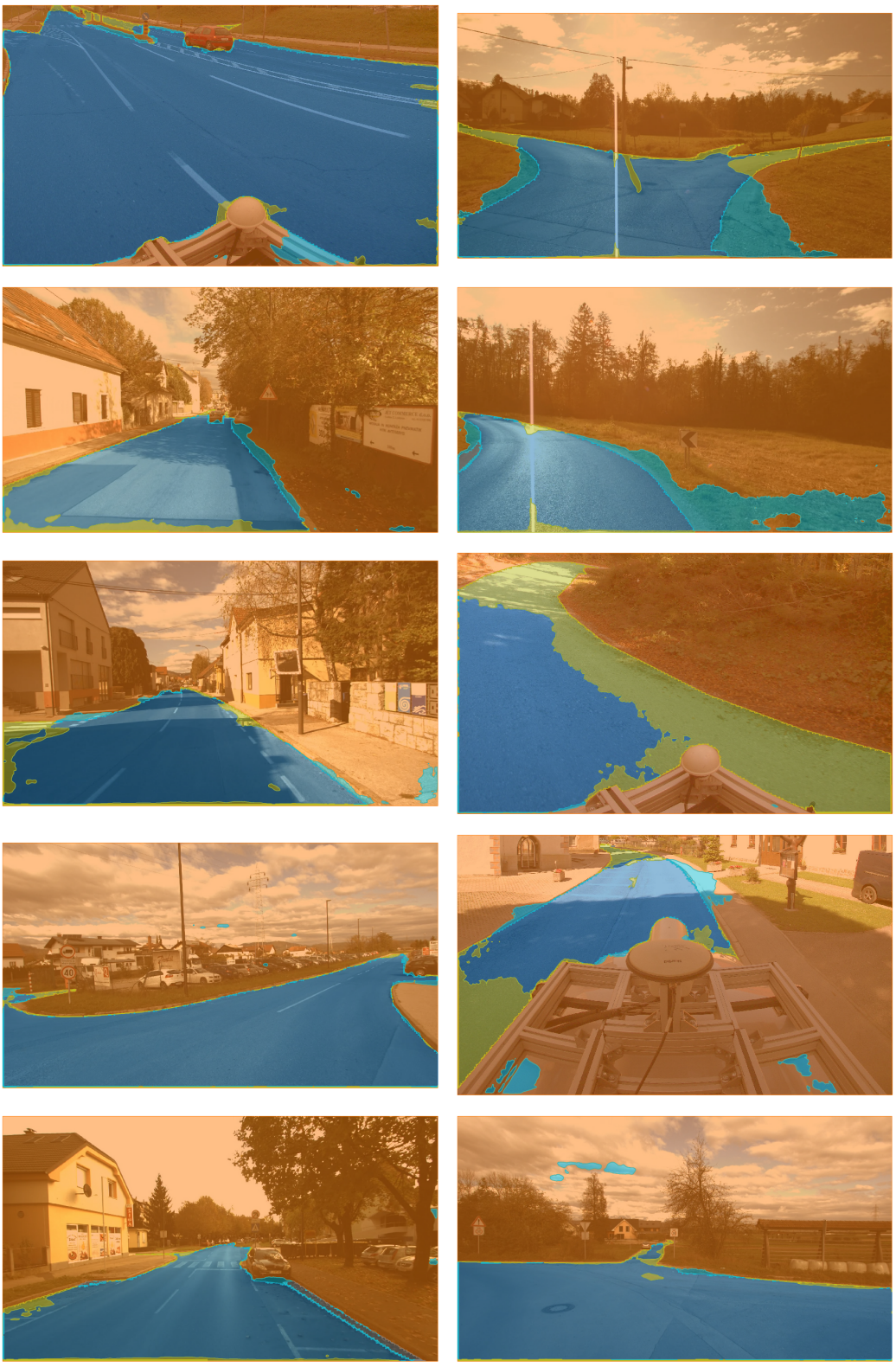
Slika 5.2 prikazuje nekaj primerov segmentacije slik zbirke namenjene klasifikaciji tipa površine cestišč (RST). Prikazana segmentacija je bila pridobljena z nevronsko mrežo naučeno na podatkovni zbirki CSF, ki je v opisanih eksperimentih pokazala najboljše rezultate. Opaziti je možno, da je uspešnost segmentacije bistveno nižja v primerjavi s Sliko 5.1, kjer so primeri vzeti iz testne množice zbirke CSF.

Da bi podobno uspešnost segmentacije dosegli na slikah zbirke RST in nato segmentacijo uporabili pri klasifikaciji tipa cestišča, smo nevronsko mrežo U-Net naučeno na zbirki CSF doučili na zbirki RSTS. Uspešnost mreže je prikazana v Tabeli 5.5. Opaziti je možno, da je uspešnost nekoliko nižja od predhodno opisanih mrež. Razlog za to se najverjetneje skriva o sami zbirki RSTS. Kot smo pokazali v poglavju 4.1.2 vsebuje zbirka RSTS nekoliko težje primere, saj so slike pogosto presvetljene, ali pa so makadamske in je določanje robov cestišča težka naloga že za človeško oko. Iz tabele je prav tako možno razbrati, da smo v primerjavi z mrežo naučeno le na zbirki RSTS, s tehniko prenosa znanja in doučenjem mreže U-Net, ki je bila predhodno naučena na zbirki CSF, izboljšali uspešnost na testni množici zbirke RSTS.

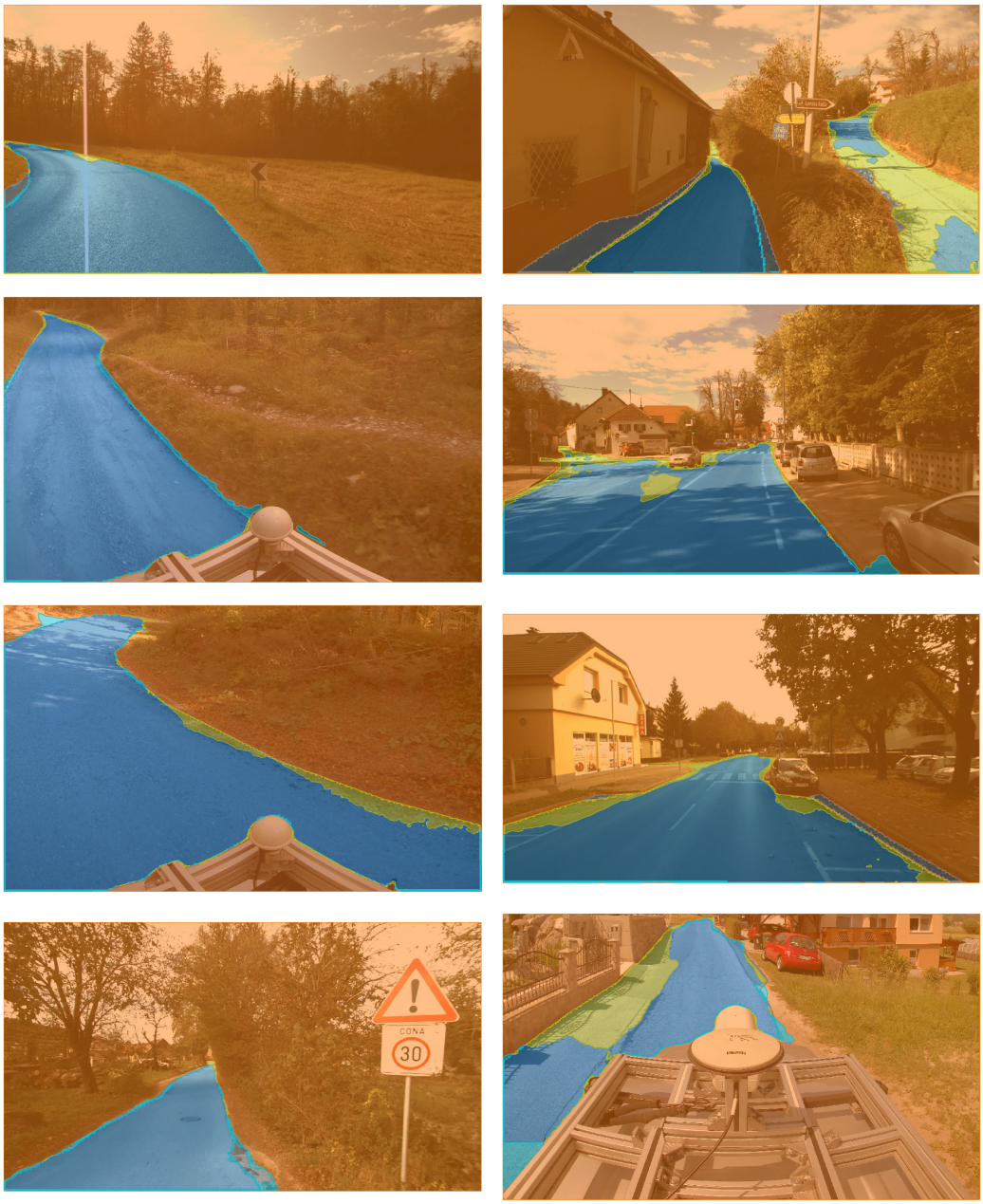
Nekaj primerov segmentacije mreže, naučene na zbirki RSTS, je možno videti na Sliki 5.3.

5.2 Klasifikacija

Za vse nadaljnje eksperimente smo v času učenja uporabili isto tehniko umetnega bogatenja učne množice in za evalvacijo isti postopek, kot smo ju opisali v poglavju 5.1, le da smo tu uporabili zbirko RTS, opisano v poglavju 4.1.1.



Slika 5.2: Prikaz nekaj primerov uspešne (levi stolpec) in neuspešne (desni stolpec) segmentacije slik zbirke RST, kjer smo za segmentacijo uporabili mrežo naučeno na zbirki CSE.



Slika 5.3: Prikaz nekaj primerov uspešne (levi stolpec) in neuspešne (desni stolpec) segmentacije slik zbirke RST, kjer smo za segmentacijo uporabili mrežo naučeno na zbirki CSE in doučeno na zbirki RSTS.

Zbirka	Preciznost	Priklic	F1
CSF	0.7220	0.8174	0.7433
RSTS	0.9428	0.9481	0.9396
CSF + RSTS	**0.9428**	**0.9608**	**0.9450**

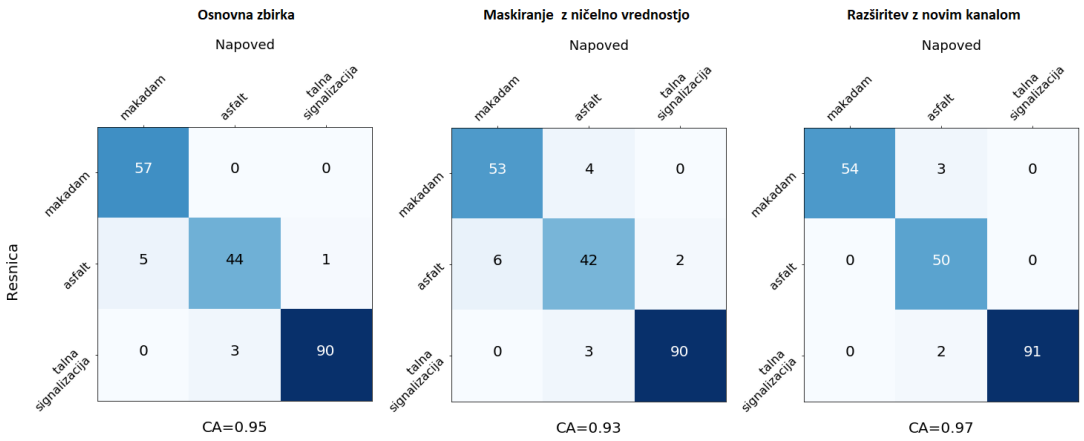
Tabela 5.5: Prikaz uspešnosti nevronske mreža arhitekture U-Net, naučene na podatkovni zbirki CSF (prva vrstica), RSTS (druga vrstica) in RSTS s predhodnim učenjem na zbirki CSF (tretja vrstica).

Mehanizem usmerjanja pozornosti	Preciznost	Priklic	F1
Brez	0.9481	0.9490	0.9477
Maskiranje z ničelno vrednostjo	0.9111	0.9123	0.9116
Razširitev z novim kanalom	**0.9697**	**0.9751**	**0.9714**

Tabela 5.6: Uspešnost klasifikacije brez dodatne informacije o položaju cestišča in rezultati z uporabo pristopov opisanih v prejšnjem poglavju.

V Tabeli 5.6 si je možno ogledati rezultate klasifikacije, ki smo jih pridobili z različnimi mehanizmi usmerjanja pozornosti. Primerjati jih je možno z rezultati klasifikacije, ki ne uporablja nikakršne informacije o položaju cestišča. Opazimo lahko, da je tehnika maskiranja z ničelno vrednostjo dosegla najslabši rezultat. Do tega najverjetneje pride zaradi izgube relevantnega konteksta, ki je sicer v ostalih dveh primerih prisoten. Z maskiranjem prav tako vpeljemo nove navidezne oblike, ki se jih nevronska mreža lahko nauči in zaradi njih izgubi sposobnost generaliziranja.

Ker v primeru napovedovanja tipa površine cestišča ne gre za binarni problem, si je napako klasifikacije smiselno pogledati s pomočjo matrik zamenjav, prikazanim na Sliki 5.4. Iz slike je možno je razbrati, da nevronska mreža, ki ne uporablja nobene dodatne informacije o položaju cestišča, najpogosteje (v 10%) zamenjuje asfaltirano cestišče z makadamskim. V 3% zamenjuje talno signalizacijo z navadnim asfaltiranim cestiščem in v 2% obratno.



Slika 5.4: Prikaz matrik zamenjav napovedi, kjer nismo uporabili informacije o položaju cestišča (levo), kjer smo uporabili tehniko maskiranja z ničelno vrednostjo (sredina) in kjer smo uporabili tehniko z razširitvijo kanalov vhodne slike (desno).

Tudi nevronska mreža, ki uporablja tehniko maskiranja vhodne slike, kjer segmentacijska mreža ni detektirala cestišča, ima v glavnem težavo z napovedovanjem asfaltiranih cestišč, ki jih najpogosteje zamenja za makadam - v 12%. Za razliko od nevronske mreže, ki ne uporablja nobene informacije o položaju cestišča, ima mreža s predhodnim maskiranjem z ničelno vrednostjo nekoliko več težav tudi z napovedovanjem makadamske ceste, ki jo pogosto zamenja za asfaltirano cestišče. V našem primeru se to zgodi v 7% vseh primerov. To se nekako sovpada z razmislekom o globalnem kontekstu, ki ga izgubimo z maskiranjem ozadja. Prav tako zamenjuje asfaltirano cestišče s cestiščem, na katerem je prisotna talna signalizacija, v 4% in v 3% zamenjuje razreda v obratni smeri.

Najboljše rezultate smo pridobili s tehniko razširitve vhodne slike z novim kanalom, ki vsebuje informacijo o položaju vozišča. Iz matrike zamenjav je možno razbrati, da je mreža v 5% zamenjala makadamsko cestišče za

asfaltirano. V 2% na asfaltiranem cestišču ni prepoznala talne signalizacije in je cestišče zato označila za asfaltirano. Nekaj uspešnih in neuspešnih primerov si je možno ogledati na Sliki 5.5.

Na tem mestu je potrebno poudariti, da tudi pri 100% uspešnosti ne smemo pričakovati, da se nevronska mreža v praksi na podobnih primerih ne bi motila. Naša testna množica je namreč vsebovala le 250 primerov, kot je to opisano v poglavju 4.1.1. Kljub temu pa je uspešnost pristopa z uporabo informacije o položaju cestišča kot četrtega kanala vhodne slike nedvomno boljša od preostalih opisanih rešitev.

Slika 5.5: Prikaz nekaj primerov uspešne in neuspešne klasifikacije.

Poglavje 6

Sklepne ugotovitve

V delu smo pokazali, kako pristopiti k problemu avtomatičnega določanja tipa površine vozišča. Pokazali smo, kako uporabiti nevronsko mrežo ResNet-50 in prenos znanja, da nekoliko zmanjšamo potrebo po ogromni količini podatkov in dolgotrajnem procesu ročnega označevanja.

Da bi klasifikacijsko uspešnost izboljšali, smo si pomagali z informacijo, kje na sliki se nahaja cestišče. Pokazali smo, v čem se razlikujejo segmentacijske nevronske mreže od klasifikacijskih, kako natančnost segmentacije izboljšamo z uporabo razširjene konvolucije in kako naučiti mrežo U-Net na podatkovni zbirki Cityscape.

Ker je večina slik v zbirki Cityscape grobo označenih, nas je zanimalo, ali lahko z uporabo pomanjkljivih oznak dosežemo primerljivo uspešnost segmentacije, kot v primeru natančno označenih podatkov. Da smo lahko uporabili neoznačene dele slik, smo si pomagali s transformacijo razdalje in predpostavko, da imajo neoznačeni slikovni elementi verjetno isti razred kot najbližje označeni slikovni element. Raziskava je pokazala, da z uporabo neoznačenih delov in različnim uteževanjem teh delov lahko izboljšamo segmentacijsko točnost, vendar pa se kljub temu ne približamo natančnosti, ki jo pridobimo z uporabo natančno označenih podatkov, četudi jih je manj.

Posledično smo se odločili za uporabo natančno označenih podatkov in natančnost poiskusili izboljšati z uteževanjem slikovnih elementov, ki se na-

hajajo na robovih cestišča, kjer se pri segmentaciji napaka največkrat pojavi. Eksperimenti so pokazali, da z uteževanjem obrobnega pasu nekoliko izboljšamo uspešnost, vendar pa ne bistveno.

Ker so slike zbirke Cityscape precej drugačne od tistih, ki smo jih uporabili za klasifikacijo tipa površine cestišča, smo nekaj takim slikam ročno ustvarili segmentacijske maske, uporabili prenos znanja in mrežo U-Net doučili tako, da je bila primerna tudi za slike, namenjene klasifikaciji. S tem modelom smo nato za vsako vhodno sliko pridobili informacijo o tem, kje na njej se nahaja cestišče. To informacijo smo uporabili za usmerjanje pozornosti klasifikacijske nevronske mreže ResNet-50.

Opravili smo dva eksperimenta in rezultate primerjali z uspešnostjo mreže, ki v času učenja ni imela dostopa do informacije o položaju cestišča. Prvi eksperiment je pokazal, da maskiranje vhodnih slik z ničelno vrednostjo uspešnost celo nekoliko poslabša. Predvidevamo, da je glavni razlog za to, izguba konteksta, ki ga nevronska mreža uporabi v nekaterih primerih. V drugem eksperimentu, smo izhod segmentacijske mreže vhodni sliki dodali kot četrti kanal. Rezultati so v tem primeru pokazali, da se je uporaba informacije o položaju cestišča izplačala, saj se je mera F1 na testni množici povečala iz 0,9477 na 0,9714.

Kljub temu da na področju semantične segmentacije z uteževanjem robov cestišča in uporabe pomanjkljivih temeljnih resnic nismo dosegli bistvenih izboljšav, smo uspeli izboljšati klasifikacijsko uspešnost napovedovanja tipa površine cestišča.

6.1 Nadaljnje delo

Kljub temu da sta si problema napovedovanja tipa in kvalitete cestišča zelo podobna, je določanje kvalitete verjetno nekoliko bolj zapleteno. Težava, s katero bi se najverjetneje soočili, je zamenjevanje senc s poškodbami. Poleg dejanskega napovedovanja indeksa MSI, ki bi bil za vzdrževalce bolj uporaben, pa bi delu lahko sledilo še nekaj zanimivih raziskav, za katere predvide-

vamo, da bi lahko izboljšale natančnost napovedovanja tipa cestišča oziroma indeksa MSI. Ker vzdrževalci običajno zajemajo slike cestišč periodično s pomočjo kamer, nameščenih na vozilih, se zaporedne slike med seboj ne razlikujejo bistveno. Posledično lahko predpostavimo, da se tudi položaj cestišča v dveh ali več zaporednih slikah ne razlikuje bistveno. S to predpostavko bi lahko uporabili rekurenčne nevronske mreže, ki bi tako za napovedovanje uporabile še časovno oziroma prostorsko dimenzijo. Podobno bi lahko predpostavili, da se v nekaj zaporednih slikah bistveno ne razlikuje tip cestišča oziroma njegova kvaliteta. Taka raziskava bi bila zanimiva z vidika analize, ali lahko z rekurenčnimi mrežami zmanjšamo zamenjavo senc s poškodbami in ali uporaba sekvenčnih podatkov privede do bolj robustnega ocenjevanja tipa oziroma kvalitete vozišča.

Literatura

[1] DFG CONSULTING, d.o.o., http://www.dfgcon.si, accessed: 29.05.2019.

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 3213–3223.

[3] A. Bearman, O. Russakovsky, V. Ferrari, L. Fei-Fei, What's the point: Semantic segmentation with point supervision, in: European Conference on Computer Vision, Springer, 2016, pp. 549–565.

[4] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, in: Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[5] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, A. Lopez, The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes, in: Comp. Vis. Patt. Recognition, 2016.

[6] S. Tsutsui, T. Kerola, S. Saito, Distantly supervised road segmentation, in: The IEEE International Conference on Computer Vision (ICCV), 2017.

[7] G. Cheng, Y. Qian, J. H. Elder, Fusing geometry and appearance for road segmentation, in: The IEEE International Conference on Computer Vision (ICCV), 2017.

[8] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Computer Vision and Pattern Recognition (CVPR), 2015.

[10] G. Huang, Z. Liu, L. van der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[11] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[12] G. Lin, A. Milan, C. Shen, I. Reid, RefineNet: Multi-path refinement networks for high-resolution semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17), 2017, [https://github.com/DrSleep/light-weight-refinenet Light-weight RefineNet with Pytorch code]. arXiv:1611.06612.

[13] O. Ronneberger, P.Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention (MICCAI), Vol. 9351 of LNCS, Springer, 2015, pp. 234–241.

[14] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, in: ICLR, 2016.

[15] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: ECCV, 2018.

[16] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[17] L. Some, Automatic image-based road crack detection methods, Master's thesis, KTH, Geodesy and Satellite Positioning (2016).

[18] H. Oliveira, P. L. Correia, Crackit — an image processing toolbox for crack detection and characterization, in: 2014 IEEE International Conference on Image Processing (ICIP), 2014, pp. 798–802. doi:10.1109/ICIP.2014.7025160.

[19] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, 2015, pp. 448–456.
URL http://jmlr.org/proceedings/papers/v37/ioffe15.html

[20] E. Shelhamer, J. Long, T. Darrell, Fully convolutional networks for semantic segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (4) (2017) 640–651. doi:10.1109/TPAMI.2016.2572683.

[21] V. Dumoulin, F. Visin, A guide to convolution arithmetic for deep learning (2016).

[22] W. Luo, Y. Li, R. Urtasun, R. Zemel, Understanding the effective receptive field in deep convolutional neural networks, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16, Curran Associates Inc., USA, 2016, pp. 4905–4913.
URL http://dl.acm.org/citation.cfm?id=3157382.3157645

[23] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.

[24] F. Yu, V. Koltun, T. A. Funkhouser, Dilated residual networks, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 636–644.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, in: CVPR09, 2009.

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 740–755.

[27] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks?, in: Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 27, Curran Associates, Inc., 2014, pp. 3320–3328.

[28] A. Odena, V. Dumoulin, C. Olah, Deconvolution and checkerboard artifacts, Distilldoi:10.23915/distill.00003.
URL http://distill.pub/2016/deconv-checkerboard

[29] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation., CoRR abs/1505.04597.

[30] O. Cuisenaire, Distance transformations: fast algorithms and applications to medical image processing, Tech. rep. (1999).

[31] L. Perez, J. Wang, The effectiveness of data augmentation in image classification using deep learning, CoRR abs/1712.04621.

[32] L. A. Gatys, A. S. Ecker, M. Bethge, A neural algorithm of artistic style (2015).

[33] Example of neural style transfer, `https://www.engadget.com/2018/12/03/nvidia-ai-video-to-video-synthesis/`, accessed on: 24.03.2019.