SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2596

NENADZIRANO UČENJE MODELA ZA MONOKULARNU PROCJENU DUBINE

Ivan Bilić

Zagreb, lipanj 2021.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2596

NENADZIRANO UČENJE MODELA ZA MONOKULARNU PROCJENU DUBINE

Ivan Bilić

Zagreb, lipanj 2021.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 12. ožujka 2021.

DIPLOMSKI ZADATAK br. 2596

Pristupnik:	**Ivan Bilić (0036495222)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	prof. dr. sc. Siniša Šegvić

Zadatak: **Nenadzirano učenje modela za monokularnu procjenu dubine**

Opis zadatka:

Monokularna procjena dubine neriješen je problem računalnog vida s mnogim zanimljivim primjenama. U posljednje vrijeme najbolja rješenja postižu se dubokim konvolucijskim modelima. Posebno su zanimljivi nenadzirani pristupi kod kojih učenje provodimo na neoznačenoj monokularnoj snimci. U okviru rada, potrebno je proučiti elemente konvolucijskih modela za monokularnu rekonstrukciju. Oblikovati odgovarajući model i naučiti ga na javno dostupnim skupovima. Validirati hiperparametre, prikazati i ocijeniti ostvarene rezultate te provesti usporedbu s rezultatima iz literature. Predložiti pravce budućeg razvoja. Radu priložiti izvorni kod razvijenih postupaka uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 28. lipnja 2021.

Posebno zahvaljujem mentoru, prof. dr. sc. Siniši Šegviću na svim savjetima i zanimljivim raspravama tijekom i prije izrade ovog rada. Također zahvaljujem asistentu Marinu Kačanu.

SADRŽAJ

1. Uvod

Čovjek je vrlo uspješan u razumijevanju trodimenzionalnog svijeta koji ga okružuje na temelju vizualnih informacija. Primjerice, hodajući ulicom čovjeku je jednostavno locirati i prepoznati objekte te prilagoditi svoje gibanje njihovom položaju. Da nije tako, često bismo razbili čašu kada bismo ju htjeli zahvatiti rukom, ali to se ne događa toliko često jer jednostavno dobro procjenjujemo udaljenost čaše od nas samih. Jedan od razloga čovjekove uspješnosti krije se u činjenici da posjeduje dva oka zbog čega u svakom trenutku vidi dvije slike i što mu omogućuje da percipira dubinu. Čovjek nije aktivno svjestan da vidi dvije slike jer ih mozak sjedinjuje, ali te dvije slike su različite, u što se lako uvjeriti sklapanjem jednog pa drugog oka naizmjence tijekom promatranja neke scene. Što je objekt od interesa ili scena općenito bliže, to je razlika između pojedinih slika veća, a što je dalje, razlika u slikama je manja. Razlika se u ovom kontekstu odnosi na činjenicu da se "pikseli" u slikama ne poklapaju. Postoje dijelovi scene koji su vidljivi za oba oka, ali se nalaze na različitim pozicijama u slici, odnosno postoji pomak. S druge strane, postoje i dijelovi scene koji su jednom oku vidljivi, a drugom nisu i obrnuto. Ta „razlika" između dvije slike koje prikazuju istu scenu naziva se disparitet i ona je vrlo važna za pouzdanu percepciju dubine, no za piksele jedne slike koji nisu vidljivi u drugoj - disparitet nema smisla, budući da za takve piksele nema korespondencija. Ako se promatra neki objekt, primjetit će se da je disparitet veći što je objekt bliže, a da je manji što je objekt dalje. Disparitet i dubina su, prema tome, obrnuto proporcionalni. Naravno, činjenica da čovjek posjeduje dva oka (i mozak čiji je rad prilagođen toj činjenici) nije jedina zaslužna za njegovu sposobnost percipiranja dubine. Dokaz tome vidljiv je ako se jedno oko sklopi - percepcija dubine ne nestaje, čini nam se da i dalje kvalitetno percipiramo dubinu, ali taj osjećaj proizlazi iz sasvim drugog razloga – apriornog iskustva i znanja o svijetu. Može se hipotetizirati da je čovjek razvio bogato strukturalno razumijevanje svijeta kroz sva vizualna iskustva koja posjeduje, a koja se naprosto sastoje od gibanja kroz svijet i promatranja ogromnog broja scena koje se u njemu nalaze (poput kamere koja se giba i snima) te razvoja konzistentnog modela vlastitih opažanja. Kroz milijune takvih opažanja, čovjek

je naučio o raznim obrascima i pravilnostima koje se u svijetu pojavljuju. Primjerice, da je kućni namještaj uglavnom određenih geometrijskih oblika i veličina, da se vozila uglavnom voze po cestama, da su ceste ravne, itd. Svo to znanje čovjek podsvjesno primjenjuje prilikom percipiranja svake nove scene. Geometrijski gledano, da bi se utvrdio podatak o dubini neke scene potrebne su najmanje dvije slike, odnosno jedna slika je nedovoljna jer postoji beskonačno mnogo različitih 3D scena iz kojih može nastati ista 2D slika, a monokularna procjena dubine je samim time *loše postavljen* (eng. *ill-posed*) problem. Međutim, čovjek svojim bogatim znanjem i iskustvom uspješno percipira i dubinu i svijet oko sebe i kada druge slike nema - kada je jedno oko zatvoreno. Zašto bi onda, iako se radi o loše postavljenom problemu, bilo nemoguće stvoriti sustav zasnovan na učenju i promatranju svijeta koji može odrediti dubinu iz samo jedne slike kao ulazne informacije? Osim čiste znatiželje i akademskog interesa, postoji još razloga za istraživanje ovog problema. Primjerice, monokularni sustav je jeftiniji od stereo sustava i primjena monokularnih kamera je široko rasprostranjena. Moguće je da primjene u robotici ili pojedini uređaji budu dimenzionirani na način da je nezgodno koristiti dvije kamere. Postoji mogućnost da i neke druge primjene u računalnom vidu imaju koristi od sustava za monokularnu procjenu dubine temeljenom na učenju. Također, s obzirom da disparitet opada s udaljenošću, za nepovoljno velike omjere udaljenosti i osnovice stereo slučaj efektivno degradira u monokularni (dvije slike postaju identične) pa je u takvim situacijama stereo sustav redundantan.

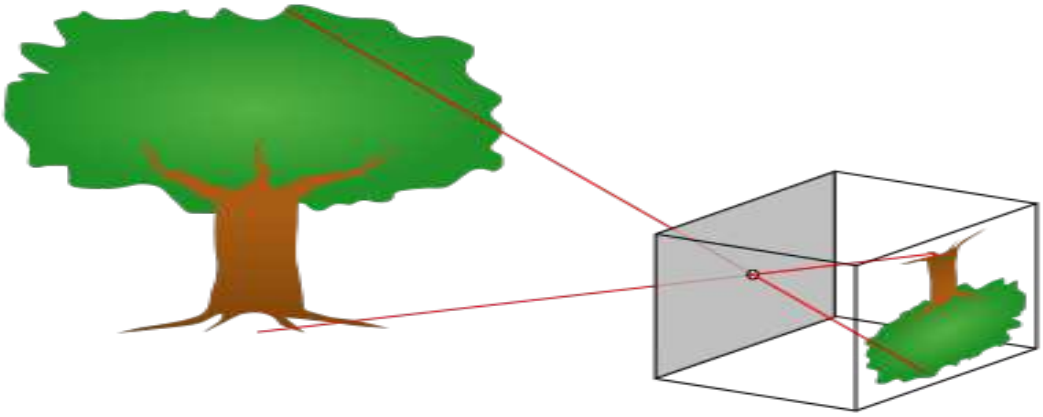
2. Geometrija nastanka slike

2.1. Model kamere

Postupak estimacije dubine iz jednog ili više pogleda tipično se provodi nad trokanalnim, RGB slikama dobivenim iz digitalne kamere. Kanali predstavljaju crvenu (R), zelenu (G) te plavu (B) boju. Intenzitet svakog piksela na slici definiran je 8-bitnim cjelobrojnim vrijednostima koje su pridružene svakom kanalu, odnosno boji. Dakle, svaki piksel je predstavljen s ukupno 24-bita jednoliko raspodijeljenih na tri boje. Kameru se može promatrati kao matematički model koji 3D točke iz svijeta preslikava u 2D točke na slikovnoj ravnini kamere. Model kamere opisuje se matricom K dimenzija 3×3 koja preslikava 3D točke svijeta u homogeni prikaz točaka u slikovnoj ravnini. Za određivanje te matrice koristi se odgovarajući kalibracijski postupak. Prilikom projekcije točke na slikovnu ravninu kamere, matrica kamere djeluje na točku kao *linearna transformacija*. Parametri kamere, poput centra projekcije, mogu se jednostavno odrediti iz matrične reprezentacije kamere. Za potrebe ovog rada pretpostavlja se perspektivni model kamere, s centralnom projekcijom i centrom koji se ne nalazi u beskonačnosti. Postoje i modeli kamere s centrom u beskonačnosti, poput primjerice *afine kamere*, koja je važna jer se paralelne linije iz svijeta preslikavaju u paralelne linije na slici, odnosno takva matrica kamere predstavlja *afinu* transformaciju radi očuvanja kolinearnosti nakon transformacije.

2.1.1. Model kamere s rupicom

Najjednostavniji model kamere je tzv. *kamera s rupicom* (eng. *pinhole camera model*). Radi se o pojednostavljenju fizičke kamere, gdje je kamera predstavljena kao kutija s malenom rupicom. Zrake svjetlosti od izvora svjetlosti padaju na objekt i osvjetljavaju ga, reflektiraju se prema kameri te prolaze kroz malenu rupicu. Nakon prolaska kroz malenu rupicu, zrake svjetlosti padaju na slikovnu ravninu, odnosno stranicu kutije nasuprot malene rupice. Radi se o modelu idealne kamere [5].



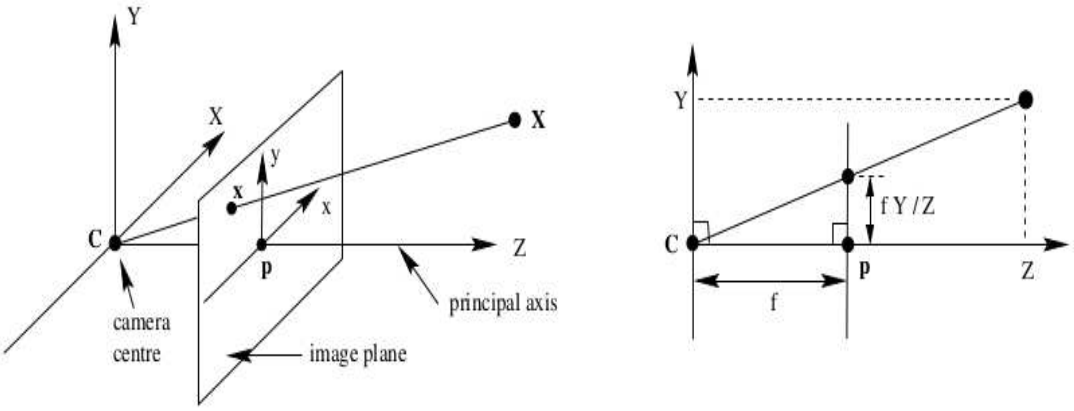
Slika 2.1: Prikaz kamere s rupicom. Zrake svjetlosti (crveni pravci) reflektiraju se od stabla, prolaze kroz rupicu te nastaje (obrnuta) slika na slikovnoj ravnini kamere. Slika je preuzeta iz [1].

Slika 2.1 prikazuje kameru s rupicom. Dva detalja su posebno zanimljiva - obrnutost slike te udaljenost malene rupice od slikovne ravnine (ravnine na kojoj nastaje slika). Udaljenost slikovne ravnine od malene rupice naziva se *žarišna duljina*. Da bi se shvatio njezin utjecaj na nastanak slike, dovoljno je zamisliti što bi se dogodilo sa slikom ako bi se žarišna duljina povećala, odnosno smanjila. Ako pogledamo sliku 2.1 i zamislimo da je udaljenost od malene rupice do slikovne ravnine veća te pretpostavimo nepromijenjenu udaljenost stvarnog objekta od rupice, jasno je da će slika objekta biti veća. Dakle, žarišna duljina određuje veličinu slike. Njezina vrijednost određuje koliki put zrake svjetlosti pređu od rupice do slikovne ravnine, a što je taj put veći, veća je i slika. Što se tiče obrnutosti slike, u svrhu pojednostavljenja geometrije slikovna ravnina može se postaviti ispred malene rupice, umjesto iza. Na slici 2.1 vidljivo je da bi zbog toga slika postala uspravna. Dokle god je žarišna duljina f ista, ispravnost modela nije narušena.

2.1.2. Geometrija modela kamere s rupicom

Neka se razmatra centralna projekcija točaka iz prostora na ravninu, neka je centar projekcije ishodište euklidskog koordinatnog sustava i neka je slikovna ravnina definirana kao $z = f$. Kod modela rupičaste kamere, točka iz prostora s koordinatama $\boldsymbol{X} = (X, Y, Z)^T$ preslikava se na slikovnu ravninu u točki $\boldsymbol{x}$ gdje pravac koji spaja točku $\boldsymbol{X}$ i centar projekcje siječe slikovnu ravninu. Navedeni pravac je zraka svjetlosti, a sve zrake svjetlosti prolaze kroz optički centar [5].

Slika 2.1 prikazuje model rupičaste kamere i geometriju nastanka slike kod tog



Slika 2.2: Geometrija modela kamere s rupicom. Točka X definirana u 3D koordinatnom sustavu kamere projicira se na 2D slikovnu ravninu kamere (lijevo). Iz bočnog prikaza (desno) proizlazi izraz koji povezuje 3D točku i njenu 2D projekciju s dubinom i žarišnom duljinom. Slika je preuzeta iz [5].

modela. (X, Y, Z) su koordinatne osi koordinatnog sustava kamere. (x, y, Z) su koordinatne osi slikovne ravnine. Z-os im je zajednička. $\boldsymbol{C}$ je centar kamere ili projekcije (malena rupica kamere), tzv. *optički centar*. Trodimenzionalni euklidski koordinatni sustav kamere postavljen je tako da je optički centar njegovo ishodište. $\boldsymbol{p}$ je *glavna točka* koja se nalazi na sjecištu osi projekcije i slikovne ravnine. Drugim riječima, $\boldsymbol{p}$ je ortogonalna projekcija ishodišta $\boldsymbol{C}$ na slikovnu ravninu. *Glavna os* je pravac koji prolazi kroz optički centar i okomit je na slikovnu ravninu. Slikovna ravnina postavljena je ispred optičkog centra i stoga je virtualna, slika nije obrnuta. Duljina dužine $\overline{\boldsymbol{Cp}}$ je vrijednost žarišne duljine f. Desni dio slike 2.2 je bočni prikaz lijevog dijela slike. Iz tog prikaza uočavaju se slični trokuti iz kojih je jednostavno izračunati da se točka $(X, Y, Z)^T$ iz svijeta preslikava u točku $(fX/Z, fY/Z, f)^T$ na slikovnoj ravnini [5].

$$f/Z = y/Y$$
$$y = fY/Z \tag{2.1}$$

Analogno vrijedi:

$$f/Z = x/X$$
$$x = fX/Z \tag{2.2}$$

Preslikavanje možemo zapisati kao:

$$(X, Y, Z)^T \mapsto (fX/Z, fY/Z, f)^T \tag{2.3}$$

Izraz (2.3) opisuje centralnu projekciju iz svjetovnih koordinata u koordinate na slikovnoj ravnini. Preslikavanje se obavlja iz euklidskog prostora $\mathbb{R}^3$ u euklidski prostor $\mathbb{R}^2$.

2.1.3. Centralna projekcija s homogenim koordinatama

Ako su točke iz svijeta i točke na slikovnoj ravnini zapisane kao homogeni vektori, centralna projekcija može se jednostavno izraziti kao linearno preslikavanje između njihovih homogenih koordinata. Homogene koordinate omogućuju elegantniji matematički zapis projekcije u odnosu na Kartezijeve koordinate. Uz danu točku (x, y) u Kartezijevim koordinatama, (xZ, yZ, Z) naziva se skupom homogenih koordinata za tu točku, $\forall Z \neq 0$. Množenjem homogenih koordinata bilo kojim skalarom različitim od nula nastaje novi skup homogenih koordinata *za istu točku*. Primjerice, točki $(2, 3)$ u Kartezijevom koordinatom sustavu odgovaraju i točka $(2, 3, 1)$ i točka $(4, 6, 2)$ u homogenim koordinatama. Drugim riječima, izvorne Kartezijeve 2D koordinate mogu se dobiti natrag iz 3D homogenih koordinata dijeljenjem prve dvije dimenzije trećom. Za gornje primjere vrijedi $x = 2/1 = 4/2 = 2$ i $y = 3/1 = 6/2 = 3$. Dakle, za razliku od Kartezijevih koordinata, jedna točka može biti predstavljena s beskonačno mnogo homogenih koordinata. Dodatno, homogenim koordinatama moguće je zapisati i točke u beskonačnosti koristeći konačne koordinate, a upravo iz tog razloga je i potrebna qwe dimenzija više za zapis točke u odnosu na Kartezijeve koordinate. Za potrebe ovog rada, važno je samo nekoliko osnovnih obilježja homogenih koordinata:

- Točki (X/Z) u 1D Kartezijevim koordinatama odgovara točka (X, Z) u 2D homogenim koordinatama ako je $Z \neq 0$. Vrijedi analogno za 2D i 3D Kartezijeve koordinate.
- Točka koju predstavljaju homogene koordinate ne mijenja se ako se sve koordinate pomnože istim faktorom različitim od nula.
- Kada je $Z = 0$, točka koja je predstavljena homogenim koordinatama nalazi se u beskonačnosti.
- $(0, 0, 0)$ u homogenim koordinatama ne odgovara nijednoj točki u Kartezijevim koordinatama, a ishodištu odgovara $(0, 0, 1)$.

Koristeći homogene koordinate, preslikavanje iz (2.3) može se zapisati kao ma-

trično množenje:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.4}$$

Matrica u gornjem izrazu može se zapisati i kao $[diag(f, f, 1)|0]$ gdje je $diag(f, f, 1)$ dijagonalna matrica dimenzija 3×3 kojoj se s desna nadoda nul-vektor stupac 3×1. Nadalje, projekcija točke iz svijeta $\boldsymbol{Q}$ u točku $\boldsymbol{q}$ na slikovnoj ravnini sada se može kompaktno zapisati:

$$\boldsymbol{q} = P\boldsymbol{Q} \tag{2.5}$$

U gornjoj jednadžbi, P nazivamo *matricom kamere*, koja za model kamere s rupicom uz centralnu projekciju definiran u (2.4), prema [5] glasi:

$$P = diag(f, f, 1)[I|0] \tag{2.6}$$

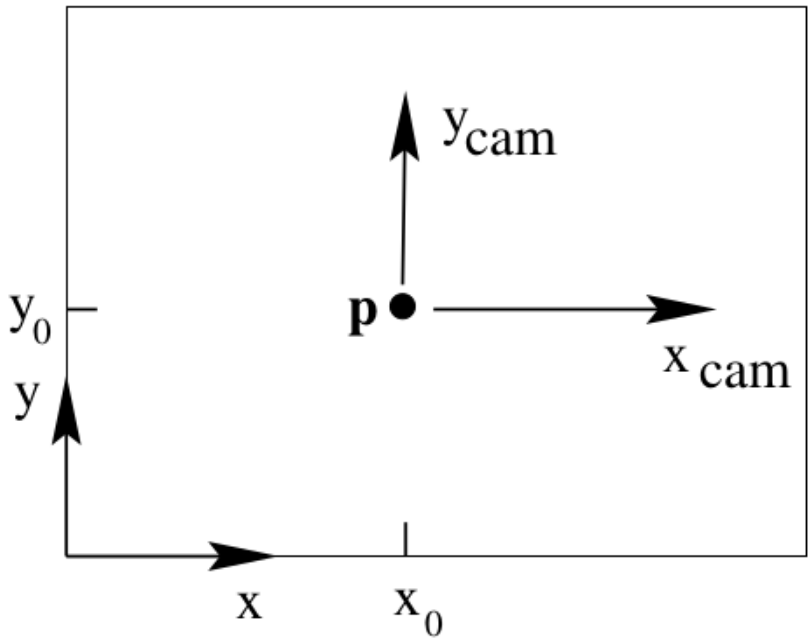
2.1.4. Pomak glavne točke

Preslikavanje iz (2.3) pretpostavlja da se ishodište koordinatnog sustava slikovne ravnine poklapa s glavnom točkom $\boldsymbol{p}$. U praksi to ne mora biti slučaj, stoga postoji općenitije preslikavanje:

$$(X, Y, Z)^T \mapsto (fX/Z + p_x, fY/Z + p_y)^T \tag{2.7}$$

gdje su $(p_x, p_y)^T$ koordinate točke $\boldsymbol{p}$. Dodavanjem tog poopćenja u (2.4) dobiva se:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX + Z\mathrm{p}_x \\ fY + Z\mathrm{p}_y \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & \mathrm{p}_x & 0 \\ 0 & f & \mathrm{p}_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.8}$$



Slika 2.3: Pomak glavne točke. Slika je preuzeta iz [5].

Pomak glavne točke prikazan je na slici 2.3. Na slici su s x i y označene koordinatne osi koordinatnog sustava slikovne ravnine, a s x_{cam} i y_{cam} koordinatne osi koordinatnog sustava kamere. Ishodište koordinatnog sustava slikovne ravnine nalazi se lijevo dolje, a ishodište koordinatnog sustava kamere poklapa se s točkom $\boldsymbol{p}$ ako se gleda duž z-osi. Pomak glavne točke $\boldsymbol{p}$ u odnosu na ishodište koordinatnog sustava slike označen je s x_0 i y_0. Drugim riječima:

$$x = x_{cam} + x_0, y = y_{cam} + y_0 \tag{2.9}$$

2.1.5. Intrinsična matrica kamere

Ukoliko se iz jednadžbe (2.8) posebno izdvoji:

$$K = \begin{bmatrix} f & & \mathrm{p}_x \\ & f & \mathrm{p}_y \\ & & 1 \end{bmatrix} \tag{2.10}$$

Konačno se dolazi do konciznog oblika jednadžbe (2.8):

$$\boldsymbol{q} = K[I|0]\boldsymbol{Q}_{cam} \tag{2.11}$$

Matrica K naziva se *intrinsična matrica kamere* [5]. Dobiva se postupkom kalibracije kamere, a žarišne duljine mogu se aproksimirati pomoću podataka koje daje proizvođač kamere. U jednadžbi (2.11) homogeni vektor $(X, Y, Z, 1)^T$ iz svijeta označen je

s $\boldsymbol{Q}_{cam}$ jer je dosad uvijek pretpostavljano da se ishodište Kartezijevog koordinatnog sustava u kojem je točka $\boldsymbol{Q}_{cam}$ definirana nalazi u optičkom centru kamere te da je navedeni koordinatni sustav okrenut tako da glavna os kamere "gleda" u smjeru njegove z-osi. Ako je točka iz svijeta definirana na takav način, odnosno ako je $\boldsymbol{Q} = \boldsymbol{Q}_{cam}$, tada se projekcija navedene točke svodi na množenje njenih homogenih koordinata s intrinsičnom matricom kamere K i zato matricu K vrijedi posebno izdvojiti [5].

Dosad razmatrani model rupičaste kamere tretirao je obje koordinatne osi slikovne ravnine s jednakim skalama. Kod realnih kamera, moguće je da su pikseli nekvadratni. Ako su koordinate na slikovnoj ravnini definirane u pikselima, tada je moguće da u različitim smjerovima vrijede različite skale. Primjerice, duljina koju predstavlja jedan piksel u smjeru x-osi, u metrima je jednaka duljini koju predstavljaju dva piksela u smjeru y-osi. Kvadrat stvarnog prostora preslikava se na "podsliku" rezolucije 1×2, umjesto samo na jedan piksel. Konkretno, ako je broj piksela po jedinici duljine na slikovnoj ravnini označen s m_x i m_y u x i y smjerovima, onda se intrinsična matrica iz (2.10) dobiva množenjem faktorom $diag(m_x, m_y, 1)$ s lijeva te nastaje općenitiji oblik intrinsične matrice:

$$K = \begin{bmatrix} \alpha_x & & \mathrm{x}_0 \\ & \alpha_y & \mathrm{y}_0 \\ & & 1 \end{bmatrix} \tag{2.12}$$

U gornjoj matrici, $\alpha_x = fm_x$ i $\alpha_y = fm_y$ predstavljaju žarišne duljine kamere definirane u prostoru piksela u x i y smjerovima. Analogno, $x_0 = m_x p_x$ i $y_0 = m_y p_y$ predstavljaju točku $\boldsymbol{p}$ definiranu u prostoru piksela.

2.1.6. Ekstrinsična matrica kamere

Ukoliko točka $\boldsymbol{Q}$ iz svijeta nije definirana u koordinatnom sustavu kamere čija projekcija se promatra, postaje jasno da će srednji član $[I|0]$ iz jednadžbe (2.11) poprimiti drugačiju vrijednost. Općenito, točke u prostoru mogu biti definirane u okviru proizvoljnog koordinatnog sustava. Ako je to slučaj, za izračun projekcije točke na slikovnu ravninu kamere potrebno je prvo napraviti transformaciju proizvoljnog koordinatnog sustava u koordinatni sustav kamere, odnosno svesti $\boldsymbol{Q}$ na $\boldsymbol{Q}_{cam}$ prema analogiji iz prethodnog potpoglavlja. Dva koordinatna sustava povezana su transformacijama rotacije i translacije. Ako je $\boldsymbol{Q}$ točka definirana u koordinatnom sustavu svijeta, prema [5], u koordinatni sustav kamere može se transformirati na sljedeći način:

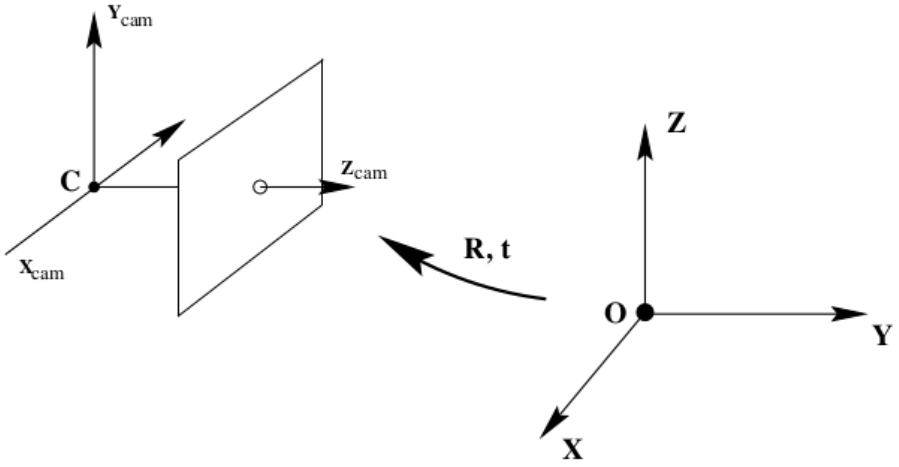
$$\widetilde{\boldsymbol{Q}}_{cam} = R(\widetilde{\boldsymbol{Q}} - \widetilde{C}) \tag{2.13}$$

$\widetilde{\boldsymbol{Q}}_{cam}$ i $\widetilde{\boldsymbol{Q}}$ su nehomogene reprezentacije u koordinatnom sustavu kamere, odnosno koordinatnom sustavu svijeta. $\widetilde{C}$ predstavlja koordinate centra kamere, tj. optičkog centra, definirane u koordinatnom sustavu svijeta. Naravno, ako je koordinatni sustav svijeta jednak koordinatnom sustavu kamere, onda je $\widetilde{C}$ nul-vektor. Dakle, $\widetilde{C}$ je dimenzija 3×1 i opisuje *translaciju* između koordinatnih sustava. R je *matrica rotacije* dimenzija 3×3 koja predstavlja orijentaciju koordinatnog sustava kamere. Drugim riječima, ako je točka definirana u koordinatnom sustavu svijeta, parametri translacije pomiču ishodište tog koordinatnog sustava u ishodište koordinatnog sustava kamere, tj. optički centar, a zatim parametri rotacije "okrenu" taj translatirani koordinatni sustav na način da se njegove osi nakon rotacije u potpunosti poklapaju s osima koordinatnog sustava kamere. Korištenjem homogenih koordinata u jednadžbi (2.13), dobiva se:

$$\boldsymbol{Q}_{cam} = \begin{bmatrix} R & -R\widetilde{C} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} R & -R\widetilde{C} \\ 0 & 1 \end{bmatrix} \boldsymbol{Q} \tag{2.14}$$

Nadalje, ako se (2.14) uvrsti u (2.11), sređivanjem nastaje:

$$\boldsymbol{q} = KR[I| - \widetilde{C}]\boldsymbol{Q} \tag{2.15}$$



Slika 2.4: Prikaz transformacije koordinatnog sustava. Koordinatni sustav svijeta (desno) se rotacijom i translacijom transformira u koordinatni sustav kamere (lijevo). Slika je preuzeta iz [5].

Slika 2.4 prikazuje općeniti slučaj preslikavanja točke iz proizvoljnog koordinatnog sustava koristeći model kamere s rupicom. Točka od interesa definirana je u ko-

ordinatnom sustavu svijeta koji se nalazi na desnom dijelu slike. Koordinatni sustav kamere i slikovna ravnina nalaze se na lijevom dijelu slike. Jednadžba (2.15) opisuje preslikavanje točke $\boldsymbol{Q}$ iz proizvoljnog koordinatnog sustava u točku $\boldsymbol{q}$ na slikovnoj ravnini. Parametri R i $\widetilde{C}$ koji povezuju orijentaciju i položaj kamere s koordinatnim sustavom svijeta nazivaju se *ekstrinsičnim* parametrima [5]. Koristeći parametre K, R i $\widetilde{C}$ matrica kamere iz (2.6) glasi:

$$P = KR[I| - \widetilde{C}] \tag{2.16}$$

Matrica P koja predstavlja općeniti opis projekcije za model rupičaste kamere ima 9 stupnjeva slobode: 3 za matricu K (elementi f, p_x, p_y), 3 za matricu R (3 Eulerova kuta) i 3 za vektor translacije $\widetilde{C}$. Konačno, ako se transformacija između koordinatnih sustava izrazi u obliku $\widetilde{\boldsymbol{Q}}_{cam} = R\widetilde{\boldsymbol{Q}} + t$, matrica kamere postaje:

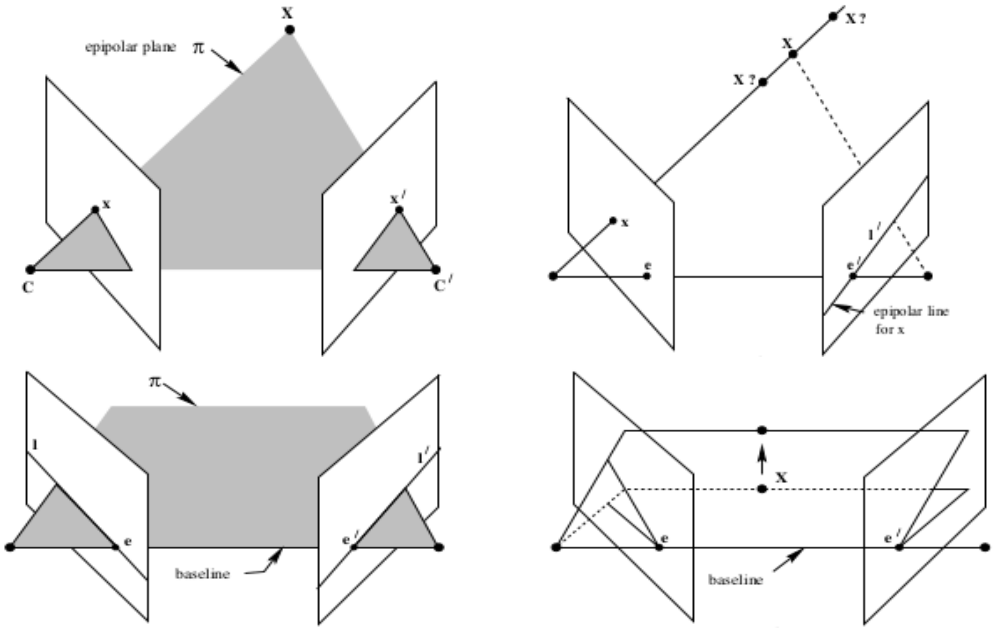
$$P = K[R|t] \tag{2.17}$$

Matrica $[R|t]$ predstavlja *ekstrinsičnu* matricu kamere, dimenzija 3×4, gdje je iz (2.16) $t = -R\widetilde{C}$.

2.2. Geometrija dvaju pogleda i izračun dubine

Određivanje dubine točaka u odnosu na poziciju kamere spada među važne zadatke u računalnom vidu. Geometrijski, estimacija dubine iz jednog pogleda (slike) je loše postavljen problem, tj. nije ju moguće odrediti, potrebna su najmanje dva. Točnije, potrebno je da se točka čija se dubina određuje pojavljuje na barem dvije slike. Ako zamislimo sustav s dvije, statične kamere usmjerene tako da su im slikovne ravnine paralelne te da su pomaknute samo po x-osi, proizvoljna točka $\boldsymbol{Q}$ koja je vidljiva na obje slike projicira se kao $\boldsymbol{q}$ na lijevu, odnosno $\boldsymbol{q}'$ na desnu slikovnu ravninu. S obzirom na to da su točke $\boldsymbol{q}$ i $\boldsymbol{q}'$ projekcije iste točke $\boldsymbol{Q}$ iz prostora, nazivaju se *korespondentne točke* ili *korespondencije* (eng. *correspondences, corresponding points*). Jedna uobičajena metoda za ekstrakciju dubine iz RGB slika svodi se na određivanje dubine iz para slika $\{I_l, I_r\}$ dobivenih koristeći dvije monokularne kamere čija je međusobna udaljenost po x-osi poznata (pretpostavimo da pomaka po y-osi nema). Takav sustav naziva se *stereo vid* (eng. *stereo vision*), a prema gornjem primjeru, horizontalno pomaknute kamere imitiraju ljudski, binokularni vid. Dakle, da bi se geometrijski odredila dubina, potrebno je pronaći korespondencije. Za pronalazak korespondencija u stereo vidu važna je *epipolarna geometrija* (eng. *epipolar geometry*). Epipolarna

geometrija opisuje međusobni položaj dvaju pogleda. Neovisna je o strukturi scene, a ovisi samo o intrinsičnim i ekstrinsičnim parametrima kamere. Epipolarna ograničenja nastaju analizom presjeka slikovnih ravnina sa skupom ravnina kandidata koje sve dijele zajedničku osnovicu - pravac koji spaja optičke centre kamera [5]. Motivacija za tu geometriju je problem pronalaska korespondentnih točaka koji se kod stereo vida rješava metodom *stereo podudaranja* (eng. *stereo matching*).

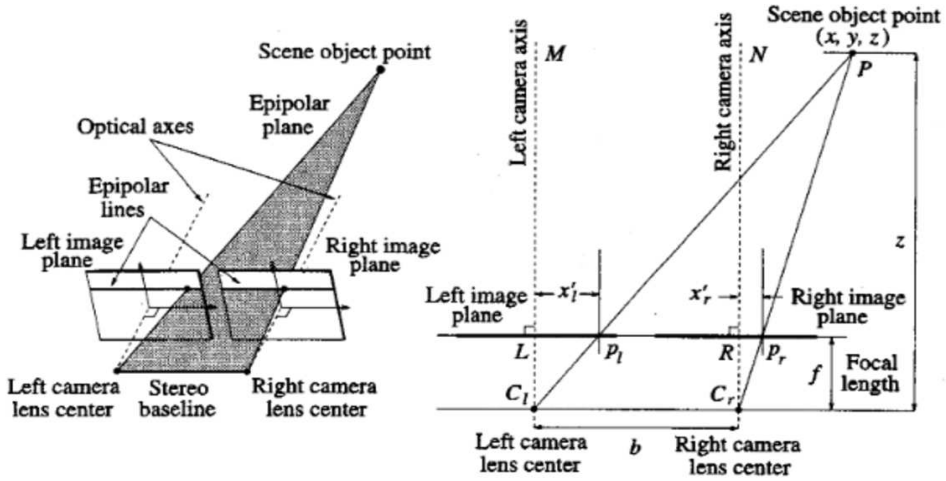


Slika 2.5: Pravci koje definiraju optički centri i 3D točka tvore jednu epipolarnu ravninu π (gore lijevo). Zraka svjetlosti koja projicira 3D točku na lijevu slikovnu ravninu, manifestira se kao epipolarna linija na desnoj slikovnoj ravnini (gore desno). Sjecišta osnovice i slikovnih ravnina su epipolovi (dolje lijevo). Promatrana 3D točka određuje epipolarnu ravninu (dolje desno). Slike su preuzete iz [5].

Gornja dva prikaza slike 2.5 prikazuju stereo vid. Točka $\boldsymbol{X}$ iz prostora se projicira na slikovne ravnine dviju kamera u $\boldsymbol{x}$ i $\boldsymbol{x}'$. $\boldsymbol{X}$ i optički centri kamera C i C' tvore epipolarnu ravninu π. Dužina $\overline{CC'}$ naziva se *osnovica*. Na gornjem desnom prikazu vidimo da ako zamislimo zraku svjetlosti iz $\boldsymbol{x}$ prema $\boldsymbol{X}$, ta se zraka manifestira kao (epipolarni) pravac l' na drugoj slici. $\boldsymbol{X}$ se mora nalaziti na toj zraki svjetlosti, stoga se i $\boldsymbol{x}'$ mora nalaziti na epipolarnoj liniji l'. Na donja dva prikaza slike 2.5 označeni su epipolovi e i e' - sjecišta osnovnice sa slikovnim ravninama. Svaka ravnina π koja sadrži osnovicu je epipolarna ravnina i siječe slikovne ravnine u odgovarajućim epipolarnim pravcima l i l'. Na donjem desnom prikazu vidi se promjena epipolarne ravnine uzrokovana promjenom položaja točke $\boldsymbol{X}$. Sve epipolarne linije sijeku se u epipolovima. Epipolarnu geometriju enkapsulira fundamentalna matrica F u slučaju

nekalibriranih kamera, odnosno esencijalna matrica E u slučaju kalibriranih kamera. Fundamentalna matrica F je stoga generalizacija esencijalne matrice E iz koje je uklonjena pretpostavka kalibriranih kamera. Prva ima 7 stupnjeva slobode, a druga 5. Korespondencije i fundamentalnu matricu povezuje $x'^T F x = 0$ gdje su x' i x homogene koordinate korespondentnih točaka. Korespondencije i esencijalnu matricu povezuje $x'^T E x = 0$ gdje su x' i x normalizirane homogene koordinate korespondentnih točaka. Normalizirane homogene koordinate dobivaju se uz pretpostavku jedinične intrinsične matrice $K = I, (f_x, f_y = 1)$ primjenom normalizirane matrice kamere $P_N = [R|t]$ koja predstavlja općenitu matricu kamere iz (2.17) pomnoženu s K^{-1} s lijeva. Formalno, fundamentalna i esencijalna matrica povezane su kroz $E = K'^T F K$, gdje su K' i K intrinsične matrice dviju kamera [5].

Uz danu projekciju $\boldsymbol{x}$ točke $\boldsymbol{X}$ na slikovnu ravninu i odgovarajuću epipolarni pravac l', proces pronalaska korespondentne točke $\boldsymbol{x}'$ svodi se na linijsko pretraživanje jer se $\boldsymbol{x}'$ mora nalaziti na odgovarajućoj epipolarnoj liniji l'. U općenitom slučaju kakav je prikazan na slici 2.5, za svaku korespondenciju potrebno je računati epipolarne linije. Situaciju je moguće dodatno pojednostaviti ako se provede postupak *rektifikacije slika*. Naime, rektifikacijom se slike transformiraju u slike koje bi nastale kada bi se slikovne ravnine dviju kamera nalazile na istoj ravnini i bile vertikalno poravnate tako da su na istim visinama. Postupku rektifikacije obično prethodi uklanjanje radijalnih izobličenja koja nastaju zbog utjecaja leće.



Slika 2.6: Izračun dubine iz geometrije dvaju pogleda. Slikovne ravnine su koplanarne, vertikalno poravnate i paralelne s osnovicom. Izraz za dubinu nastaje iz sličnih trokuta (desno). U njemu se pojavljuje disparitet - horizontalni pomak korespondencija. Slika je preuzeta iz [10].

Lijevi dio slike 2.6 prikazuje dvije jednake slikovne ravnine koje su koplanarne, istih vertikalnih pozicija te paralelne s pravcem kojeg određuju optički centri (osnovica). Jedna od posljedica takvih geometrijskih postavki je da su intrinsične matrice K dviju kamera jednake, $K = K^{'}$ jer su žarišne duljine f jednake, a glavne točke p dviju kamera imaju istu vrijednost. Kamere su pomaknute samo u smjeru x-osi, duljina pomaka je osnovica b (eng. *baseline*). U ovakvom modelu, za svaku korespondenciju vrijedi da je redak u kojem se nalazi x jednak retku u kojem se nalazi $x^{'}$ na drugoj slici (u prostoru piksela). Pronalazak korespondencija sveden je s dvodimenzionalnog na jednodimenzionalan problem, a pošto znamo da se $x^{'}$ nalazi u istom retku kao i x, epipolarne linije su poznate. Jedini pomak koji postoji između korespondencija je horizontalni pomak - *disparitet*. Disparitet je definiran razlikom x koordinata (izraženih u 2D koordinatnim sustavima slikovnih ravnina) korespondentnih točaka. Na desnom prikazu slike 2.6, točka P iz svijeta projicira se na dvije slikovne ravnine kao p_l, odnosno p_r. Neka je ishodište koordinatnog sustava postavljeno u optički centar lijeve kamere. Na slici postoje slični trokuti PMC_l i p_lLC_l:

$$\frac{x}{z} = \frac{x_l^{'}}{f} \tag{2.18}$$

Analogno, iz sličnih trokuta PNC_r i p_rRC_r:

$$\frac{x-b}{z} = \frac{x_r^{'}}{f} \tag{2.19}$$

Supstitucijom x u (2.19) pomoću (2.18):

$$z = \frac{bf}{x_l^{'} - x_r^{'}} = f\frac{b}{d} \tag{2.20}$$

Nazivnik u izrazu (2.20) je disparitet [10]. Izraz govori da se dubina može izračunati poznavajući disparitet d između korespondencija, osnovicu b između optičkih centara kamera te žarišnu duljinu f. Kroz izraz (2.20), jasno se ističe razlika između stereo i monokularne procjene dubine. S obzirom da disparitet po prirodi definicije podrazumijeva više od jednog pogleda, on kao veličina nema smisla u monokularnom slučaju. Kod monokularne procjene dubine, disparitet se dobiva predikcijom. Osim toga, monokularna i stereo procjena dubine razlikuju se i po osnovici b. Naime, osnovica je pristuna (i fiksna) samo u stereo slučaju. Kod monokularne procjene dubine, dubina se u trenutku t procjenjuje samo na temelju jednog pogleda, nema "desne" kamere pa nema ni osnovice b. Dakle, ako f i d iz (2.20) izrazimo u pikselima, a b u metrima, dubina je izražena u metrima. S druge strane, u ovome radu razmatra se monokularni model temeljen na dubokom učenju koji procjenjuje disparitet. Za izračun dubine, taj se model oslanja isključivo na obrnutu proporcionalnost dubine i dispariteta (2.20). Prema tome, navedeni model nema fizikalno potkrijepljenu mjernu jedinicu za dubinu.

3. Gradivni elementi modela za monokularnu procjenu dubine

Područje umjetne inteligencije od svojih ranih dana pokušava riješiti mnoge zanimljive probleme koji su intelektualno zahtjevni za ljude, a pogodni za računala - probleme koji se mogu opisati popisom formalnih, matematički definiranih pravila. No, pravi izazov u području umjetne inteligencije pokazao se u rješavanju problema koje ljudi lako rješavaju, ali ih teško formalno opisuju - problemi koje ljudi rješavaju intuitivno, "automatski", poput prepoznavanja izgovorenih riječi ili lica na slikama. Jedan mogući pristup za rješavanje takve vrste problema je dozvoliti računalu da uči iz iskustva te temeljiti njegovo razumijevanje svijeta na hijerarhiji koncepata, gdje je svaki koncept definiran kroz odnos s jednostavnijim konceptima. Ako se pristup temelji na učenju iz iskustva, ne postoji zahtjev da čovjek formalno opiše svo znanje koje je računalu potrebno. Hijerarhija koncepata omogućuje učenje složenijih koncepata, izgradnjom istih iz jednostavnijih. Crtanjem grafa koji pokazuje kako su ti koncepti izgrađeni jedni nad drugima, došli bismo do grafa koji je dubok, s mnogo slojeva. Iz tog razloga, navedeni pristup umjetnoj inteligenciji naziva se *duboko učenje* [4].

3.1. Konvolucijska neuronska mreža - temelji

Konvolucijske neuronske mreže (eng. convolutional neural networks, abbr. CNNs) specijalizirana su vrsta neuronskih mreža za obradu podataka koji imaju topologiju nalik rešetke. Primjerice, podaci iz vremenskih serija mogu se promatrati kao jednodimenzionalna rešetka u kojoj su elementi uzorkovani kroz vremenske intervale. Slike se mogu promatrati kao dvodimenzionalna rešetka piksela. Konvolucijske neuronske mreže pokazale su se uspješnima u praktičnim primjenama. Naziv implicira da su zasnovane na matematičkoj (linearnoj) operaciji zvanoj *konvolucija*. Konvolucijske neuronske mreže su vrsta neuronskih mreža koja koristi konvoluciju umjesto standard-

nog matričnog množenja [1] u barem jednom od svojih slojeva [4].

3.1.1. Konvolucija

Općenito, konvolucija je operacija nad dvije funkcije s realnim argumentima, s formalnim zapisom:

$$\int x(a)\omega(t-a)\,da \tag{3.1}$$

Operacija konvolucije često se označava pomoću zvjezdice:

$$s(t) = (x * \omega)(t) \tag{3.2}$$

U kontekstu konvolucijskih neuronskih mreža, s obzirom na gornje jednadžbe, funkciju x tretiramo kao ulaz, a funkciju ω zovemo *jezgrom* ili *filtrom*. Rezultat konvolucije je *mapa značajki* (eng. *feature map*). S obzirom da je kod slika prostor piksela diskretan, t može poprimiti samo cjelobrojne vrijednosti. Pod pretpostavkom da su x i ω definirani samo nad cjelobrojnim t, definira se diskretna konvolucija:

$$s(t) = (x * \omega)(t) = \sum_{a=-\infty}^{\infty} x(a)\omega(t-a) \tag{3.3}$$

U primjenama strojnog učenja, ulaz je tipično višedimenzionalni niz vrijednosti, a konvolucijski filter višedimenzionalni niz parametara koji se optimiraju na skupu za učenje. Takve višedimenzionalne nizove često se naziva *tenzorima*. Pretpostavlja se da funkcije x i ω imaju konačnu domenu. Zbog navedene prirode ulaznih podataka, konvolucija se uglavnom provodi nad više osi. Primjerice, za RGB slike na ulazu, želimo koristiti dvodimenzionalan konvolucijski filter K:

$$S(i,j) = (I * K)(i,j) = \sum_{m}\sum_{n} I(m,n)K(i-m, j-n) \tag{3.4}$$

Diskretna konvolucija može se promatrati kao matrično množenje, gdje postoje ograničenja na određene elemente matrice. Primjerice, kod jednodimenzionalne diskretne konvolucije, svaki redak matrice je ograničen na način da mora biti jednak retku iznad pomaknutome za jedan element [2]. Slično vrijedi i u dvije dimenzije. Uz ograničenja gdje neki elementi trebaju biti jednaki jedni drugima, matrica koja predstavlja konvoluciju je vrlo rijetka matrica - većina njenih elemenata je nula. Razlog leži u tome što

[1] Standardno matrično množenje odnosi se na transformaciju koju izvodi klasičan umjetni neuron, $W^T X$

[2] Poznato i kao Toeplitzova matrica. Pretpostavlja se konvolucija s pomakom 1.

je konvolucijski filter gotovo uvijek puno manjih dimenzija od ulazne slike.

$$S(\boldsymbol{X}) = \begin{pmatrix} \omega_{0,0} & \omega_{0,1} & 0 & \omega_{1,0} & \omega_{1,1} & 0 & 0 & 0 & 0 \\ 0 & \omega_{0,0} & \omega_{0,1} & 0 & \omega_{1,0} & \omega_{1,1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \omega_{0,0} & \omega_{0,1} & 0 & \omega_{1,0} & \omega_{1,1} & 0 \\ 0 & 0 & 0 & 0 & \omega_{0,0} & \omega_{0,1} & 0 & \omega_{1,0} & \omega_{1,1} \end{pmatrix} \begin{pmatrix} x_{0,0} \\ x_{0,1} \\ x_{0,2} \\ x_{1,0} \\ x_{1,1} \\ x_{1,2} \\ x_{2,0} \\ x_{2,1} \\ x_{2,2} \end{pmatrix} \tag{3.5}$$

Izraz (3.5) prikazuje 2D diskretnu konvoluciju kao matrični umnožak $\boldsymbol{C}\boldsymbol{X}_{col}$. Pritom je $\boldsymbol{C}$ matrica s težinama konvolucijskog filtra, a $\boldsymbol{X}_{col}$ je ulaz dimenzija 3×3 organiziran kao vektor-stupac. U danom izrazu, konvolucijski filter sastoji se od 4 težine. Ako bismo ga nacrtali u 2D, dobili bismo prozor dimenzija 2×2 kojeg bismo pomicali po ulazu računajući sume produkata. U izrazu je korišten pomak 1 između produkata pojedinih "prozora" (retci matrice $\boldsymbol{C}$) i ulaza $\boldsymbol{X}_{col}$. Zbog prethodno navedenih dimenzija te iznosa pomaka, diskretna konvolucija je suma 4 produkta u danom primjeru. Zbog toga matrica C ima 4 retka. Rezultat izraza (3.5) je mapa značajki dimenzija 4×1, koju se presloži u matricu 2×2. Također, u danom primjeru nije korišteno nadopunjavanje (eng.) kod kojeg se rubovi ulaza "nadopune" dodavanjem novih elemenata. Time se postignu drugačije dimenzije izlazne mape značajki, a elementi na rubovima ulaza se više iskoriste.

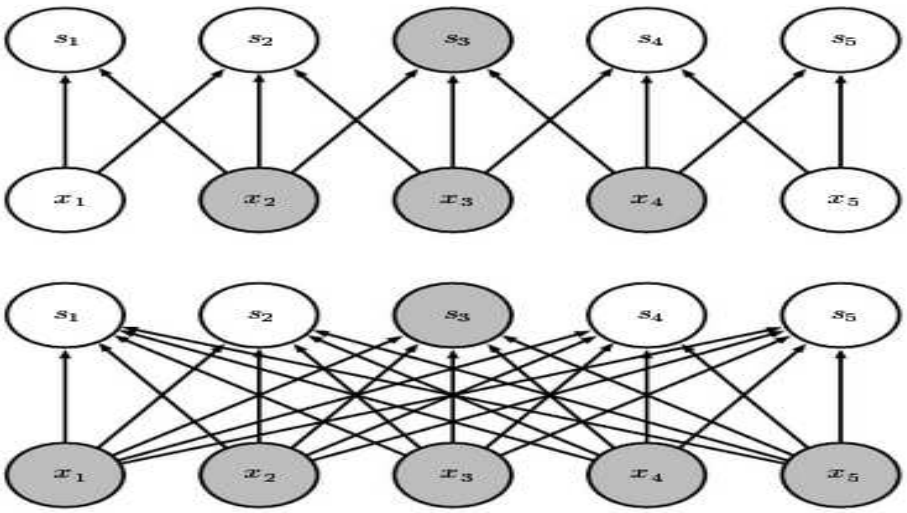
Efekti koji se manifestiraju u algoritmu dubokog učenja kao posljedica zasnivanja neuronske mreže na operaciji konvolucije su sljedeći:

- rijetka povezanost
- dijeljenje parametara
- ekvivarijantna preslikavanja

3.1.2. Rijetka povezanost

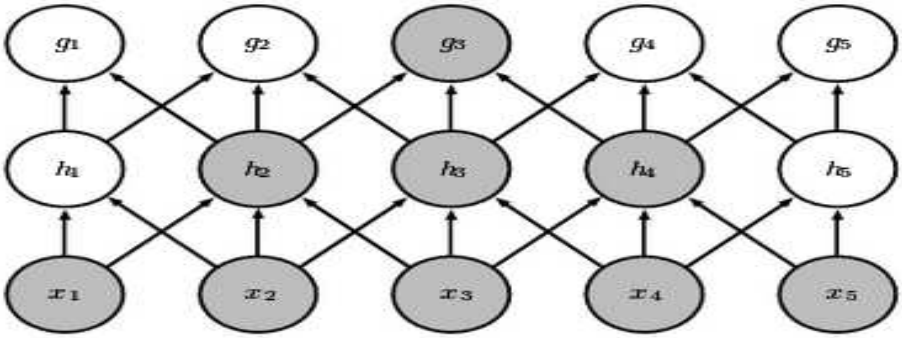
Rijetka povezanost odnosi se na broj ulaznih interakcija po izlaznoj jedinici. Primjerice, slojevi potpuno povezane unaprijedne neuronske mreže koriste matrično množenje s matricom koja sadrži parametre sloja gdje za svaku ulaznu jedinicu postoji odgovarajući parametar. Drugim riječima, svaka izlazna aktivacija ima interakciju sa

svakom ulaznom jedinicom iz prethodnog sloja (sirovi ulazni podaci su nulti sloj). Konvolucijske neuronske mreže, s druge strane, imaju rijetku povezanost koja postignuta time što su filteri manjih dimenzija od ulaza. Dobrobiti toga uključuju pohranu manjeg broja parametara, manji broj potrebnih operacija za izračun izlazne vrijednosti, a i detekcija sitnih značajki koje okupiraju manji broj piksela na ulaznoj slici.



Slika 3.1: Kod izračuna konvolucije (gore), svaka izlazna aktivacija ima interakciju samo s nekoliko ulaznih elemenata. Kod potpuno povezanih slojeva (dolje), svaka izlazna aktivacija ima interakciju sa svim ulaznim elementima. Slika je preuzeta iz [4].

Slika 3.1 prikazuje rijetku (gore) i gustu (dolje) povezanost. Ako je na gornjem dijelu slike izlaz s_3 dobiven konvolucijskim filterom širine 3, s_3 je povezan samo s ulazima x_2, x_3 i x_4, koji ujedno predstavljaju *receptivno polje* od s_3. Donji dio slike prikazuje potpunu povezanost, kod koje izlaz s_3 vidi sve ulaze x_i, što je slučaj u potpuno povezanim neuronskim mrežama. Važno je primijetiti da, unatoč rijetkoj povezanosti, u dubokim konvolucijskim mrežama dublji slojevi su indirektno povezani s većim brojem ulaza. To omogućuje mreži da modelira složene interakcije u kasnijim slojevima koristeći rijetke interakcije iz ranijih slojeva.



Slika 3.2: Receptivno polje raste od plićih prema dubljim slojevima konvolucijske neuronske mreže. Sivom bojom naglašeno je receptivno polje aktivacije g_3. Zbog hijerarhijske izgradnje značajki, kasniji slojevi detektiraju složenije značajke u ulazu. Slika je preuzeta iz [4].

Slika 3.2 prikazuje efekt da je receptivno polje dubljih slojeva veće od receptivnog polja u plićim slojevima. Taj se efekt dodatno može pojačati korištenjem arhitekturalnih dodataka poput konvolucija s većim korakom, dilatiranih konvolucija ili sažimanja. Dakle, iako je izravna povezanost u konvolucijskoj mreži rijetka, dublji slojevi mogu neizravno biti povezani s većinom ili gotovo cijelom ulaznom slikom.

3.1.3. Dijeljenje parametara i ekvivarijantnost s obzirom na pomak

Dijeljenje parametara odnosi se na korištenje istog skupa parametara za više od jedne funkcije unutar modela. Kod potpuno povezane unaprijedne neuronske mreže, svaki parametar svakog sloja koristi se samo jednom pri izračunu izlaza sloja. Pomnožen je samo jedanput s jednom ulaznom vrijednosti. Kod konvolucijske neuronske mreže, svaki parametar svakog filtera koristi se na svakoj poziciji na kojoj se filter nađe prilikom izračuna konvolucije. Dakle, umjesto da se za svaku lokaciju uči zaseban skup parametara, uči se samo jedan skup. To ne mijenja složenost vremena izvođenja unaprijedne propagacije, ali smanjuje broj parametara i otežava prenaučenost. Dodatno, pojedini filteri uče pojedine obrasce u podacima, odnosno aktiviraju se na njih. Ako je neki filter kroz postupak učenja naučio težine s kojima prepoznaje vertikalnu liniju unutar prozora dimenzija 3×3, svejedno je na kojem se dijelu slike ona nalazi, zbog dijeljenja parametara.

Dijeljenje parametara unutar slojeva konvolucijske neuronske mreže uzrokuje *ekvivarijantnost s obzirom na pomak*. Svojstvo ekvivarijantnosti funkcije znači da se izlaz funkcije mijenja identično s promjenom ulaza. Drugim riječima, ako je funkcija

$f(x)$ ekvivarijantna s obzirom na funkciju $g(x)$, onda $f(g(x)) = g(f(x))$. U kontekstu konvolucije, ako je g funkcija koja translatira ulaz, konvolucija je ekvivarijantna s obzirom na g. Konkretno, konvolucija stvara dvodimenzionalnu mapu značajki koja govori gdje se na ulaznoj slici pojavljuju određene značajke. Ako se neki objekt pomakne na neko drugo mjesto na ulaznoj slici, rezultat konvolucije nad tim objektom bit će pomaknut za isti iznos u mapi značajki. Konvolucija nije prirodno ekvivarijantna s obzirom na neke druge transformacije, poput promjene mjerila ili rotaciju slike. Drugi mehanizmi su potrebni za adresiranje takvih transformacija [4].

Konačno, zbog prirode konvolucije, konvolucijske neuronske mreže fleksibilnije su po pitanju ulaznih podataka s kakvima mogu raditi. Primjerice, na ulaz se prihvaća slika proizvoljnih dimenzija. Može se, primjerice konvoluciju koristiti i za obradu prirodnog jezika, na proizvoljno velikim nizovima tokena. To svojstvo ne vrijedi za poput potpuno povezane unaprijedne neuronske mreže koje su ograničene na ulaze fiksnih dimenzija. Kod promjene dimenzije ulaza (rezolucija slike, duljina niza tokena, itd.) matrično množenje ne bi funkcioniralo zbog nepoklapanja potrebnih dimenzija matrice s podacima i matrice parametara.

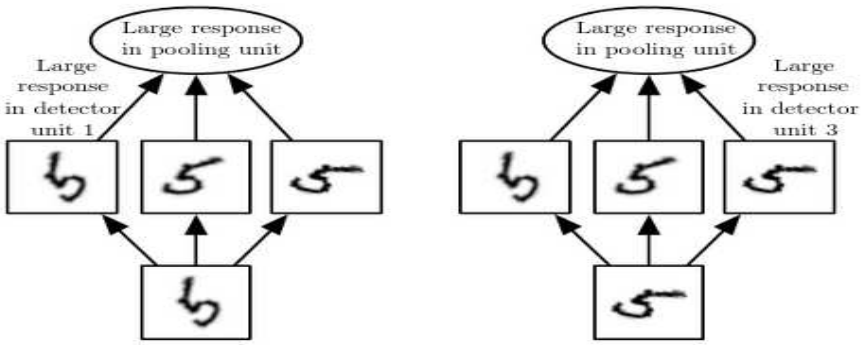
3.1.4. Sažimanje

Jedan sloj konvolucijske neuronske mreže tipično se sastoji od tri faze. U prvoj fazi računaju se konvolucije (linearne aktivacije) nad svim pozicijama za sve filtere unutar sloja, a broj novonastalih mapi značajki jednak je broju filtera unutar samog sloja. U drugoj fazi, svaka linearna aktivacija dovodi se na ulaz nelinearne aktivacijske funkcije, poput primjerice ReLU funkcije. U trećoj fazi, ako ista postoji, primjenjuje se funkcija sažimanja[3]. Funkcija sažimanja zamjenjuje izlaznu vrijednost mreže na određenoj lokaciji uzimajući statistiku okolnih izlaza u obzir. Primjerice, funkcija sažimanja maksimumom na ulaz prima pravokutni prozor vrijednosti iz mape značajki i na izlaz daje maksimalnu vrijednost koja se nalazi unutar prozora[4]. Ako je prozor dimenzija 2×2, tada će 4 izlazne vrijednosti iz mape značajki biti zamijenjene jednom, maksimalnom vrijednosti od te 4. Neke druge popularne funkcije sažimanja uključuju sažimanje prosjekom, L2 normu ili težinski prosjek na temelju udaljenosti od središnjeg piksela. Nadalje, sažimanje utječe na reprezentaciju na način da postaje invarijantna na malene translacije ulaza. Invarijantnost na translaciju znači da ako je ulaz translatiran za neki

[3]Korištenje sažimanja u svakom konvolucijskom sloju vrlo je rijetko. Tipično iza konvolucije slijedi samo nelinearnost.

[4]Prozor može biti i dimenzija 1×1, čime se sažimanje primjenjuje za svaki element zasebno.

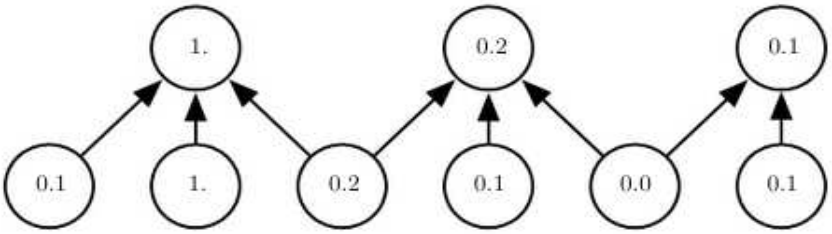
malen iznos, većina izlaznih vrijednosti sažimanja ostaje nepromijenjena. Invarijantnost na lokalne translacije može biti korisno svojstvo - ako je važnije postoji li neka značajka, nego gdje se točno nalazi. Primjerice, kada se određuje postoji li ljudsko lice na slici, nije potrebno znati točnu lokaciju piksela koje oči zauzimaju, već samo da postoje lijevo i desno oko na lijevoj i desnoj strani lica. S druge strane, postoje slučajevi gdje je znanje o lokaciji značajke važno. Primjerice, ako je potrebno pronaći ugao definiran dvama rubovima koji se na slici sastaju na određeni način, važno je da lokacija rubova bude dovoljno očuvana da bi se provjerilo sastaju li se na odgovarajući način. Korištenje sažimanja može se promatrati kao uvođenje snažne apriorne distribu-



Slika 3.3: Invarijantnost na rotaciju korištenjem sažimanja. Različiti filteri konvolucijskog sloja detektiraju različite orijentacije znamenke 5. Za svaku od tih orijentacija na ulazu, sažimanjem po dimenziji kanala dobiva se slična aktivacija. Slika je preuzeta iz [4].

cije nad parametrima modela, odnosno da funkcija koju sloj uči treba biti invarijantna na lokalne translacije. Kada je ta pretpostavka ispravna, sažimanje može imati značajan utjecaj na statističku učinkovitost mreže. Dvodimenzionalnim sažimanjem - po širini i visini jedne mape značajki, za svaku mapu značajki, postiže se invarijantnost na lokalne translacije. Međutim, ako se sažimanje provodi i nad trećom dimenzijom, po mapama značajki koje su rezultat različito parametriziranih konvolucija, može se postići efekt pri kojem mreža nauči na koje transformacije treba biti invarijantna [4]. Slika 3.3 prikazuje jedan takav primjer. Neka je na ulazu mreže jednokanalna slika znamenke 5. Mreža se sastoji od 3 konvolucijska filtera od kojih je svaki naučio aktivaciju na različitu orijentaciju znamenke. Ako se na ulaz dovede znamenka kao na lijevoj slici, najsnažnija aktivacija bit će od prvog filtera. Ako se dovede znamenka kao na desnoj slici, najsnažnija aktivacija bit će od desnog filtera. Za oba scenarija, aktivacija sažimanja maksimumom (po kanalima) bit će velika. Dakle, dva različita ulaza, aktivacije dva različita detektora, a utjecaj na sažimanje je gotovo isti u oba slučaja.

Sažimanje se može koristiti i za povećanje računalne efikasnosti, postavljanjem pomaka između susjednih operacija sažimanja na k piksela umjesto na 1, čime se postiže efekt poduzorkovanja reprezentacije. Na slici 3.4 prikazano je sažimanje maksimu-



Slika 3.4: Sažimanje s poduzorkovanjem

mom širine prozora 3 i s pomakom između susjednih sažimanja iznosa 2. Time se rezolucija reprezentacije smanjuje za faktor 2, što smanjuje računalno opterećenje sljedećeg sloja mreže. Može se dogoditi da za posljednje sažimanje ne preostane elemenata koliko stane u regiju jednog sažimanja, no sažimanje se bez obzira može izračunati, u suprotnom bi se takve rubne aktivacije ignorirale. Nadalje, kada je broj parametara nekog sloja funkcija koja ovisi o dimenzijama ulaza - primjerice potpuno povezani sloj, a model na ulaz dobiva slike promjenjivih rezolucija, sažimanje omogućuje ispravnu obradu. Primjera radi, ulaz na potputno povezani sloj (koji se često koristi kao izlazni, klasifikacijski sloj prije softmax funkcije) mora biti fiksnih dimenzija. Taj se zahtjev može zadovoljiti tako da se tijekom unaprijednog prolaska izračunaju parametri sažimanja (dimenzije regije sažimanja, pomak) koji će osigurati da klasifikacijski sloj na ulaz uvijek dobije reprezentaciju koja je dimenzija kakve očekuje, neovisno o dimenzijama ulaznih podataka. Alternativno, u modernim dubokim modelima za klasifikaciju popularnija opcija je globalno sažimanje, kod kojeg prozor za sažimanje prekriva cijelu mapu značajki.

3.2. Normalizacija grupe i problematika optimizacije

Uobičajena praksa kod algoritama dubokog učenja je da se unaprijedni prolazak podataka kroz duboku neuronsku mrežu računa nad *mini-grupom* podataka (eng. *mini-batch*). Ako je skup podataka za učenje veličine N, neka je mini-grupa veličine n, uz $n > 0$ i $n << N$. U praksi, težine neuronske mreže optimiraju se postupkom stohastičke optimizacije. Prvo se izračunaju predikcije za cijelu mini-grupu, a tek onda se ažuriraju parametri modela na temelju izračunatih gradijenata funkcije gubitka s obzi-

rom na parametre modela. Gradijent pritom nije definiran nad cijelim skupom za učenje, nego samo nad mini-grupom. To čini postupak stohastičkim. Normalizacija grupe motivirana je težinom učenja koja dolazi do izražaja kod vrlo dubokih modela koji se sastoje od većeg broja slojeva[5]. Tijekom optimizacijskog postupka, gradijent određuje kako ažurirati svaki parametar svakog sloja, pod pretpostavkom da se ostali slojevi ne mijenjaju. Međutim, u praksi se parametri svih slojeva modela ažuriraju istvoremeno[6]. Nakon ažuriranja parametara, postoji mogućnost da rezultati budu "neočekivani" jer se mnogo funkcija koje su kompozicije (unaprijedni prolazak) mijenjaju istovremeno, koristeći ažuriranja izračunata pod pretpostavkom da se ostale funkcije ne mijenjaju [4]. Efekti istovremenog ažuriranja parametara različitih slojeva modela mogu se dočarati jednostavnim primjerom iz [4]. Primjerice, neka je duboka neuronska mreža sastavljena od jednog neurona po sloju l, bez aktivacijskih funkcija u skrivenim slojevima: $\hat{y} = x\omega_1\omega_2\omega_3...\omega_l$. ω_i je parametar koji odgovara sloju i. Izlaz sloja i je onda: $h_i = h_{i-1}\omega_i$. Izlaz mreže $\hat{y}$ je linearna funkcija ulaza x, a nelinearna funkcija težina ω_i. Algoritam unazadne propagacije pogreške izračunat će gradijent $\boldsymbol{g} = \nabla_\omega \hat{y}$. Neka je pravilo ažuriranja parametara modela: $\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \epsilon \boldsymbol{g}$. Aproksimacija funkcije gubitka $f(\boldsymbol{\omega})$ razvojem u Taylorov red drugog stupnja u okolini točke $\boldsymbol{\omega}^{(0)}$ glasi:

$$f(\boldsymbol{\omega}) \approx f(\boldsymbol{\omega}^{(0)}) + (\boldsymbol{\omega} - \boldsymbol{\omega}^{(0)})^T \boldsymbol{g} + \frac{1}{2}(\boldsymbol{\omega} - \boldsymbol{\omega}^{(0)})^T \boldsymbol{H} (\boldsymbol{\omega} - \boldsymbol{\omega}^{(0)}) \tag{3.6}$$

$\boldsymbol{g}$ je gradijent, a $\boldsymbol{H}$ Hesseova matrica u $\boldsymbol{\omega}^{(0)}$. Nadalje, koristeći stopu učenja ϵ pri ažuriranju parametara, novi parametri $\boldsymbol{\omega}$ dani su s $\boldsymbol{\omega}^{(0)} - \epsilon \boldsymbol{g}$. Uvrštavanjem pravila ažuriranja u (3.6):

$$f(\boldsymbol{\omega}^{(0)} - \epsilon \boldsymbol{g}) \approx f(\boldsymbol{\omega}^{(0)}) - \epsilon \boldsymbol{g}^T \boldsymbol{g} + \frac{1}{2}\epsilon^2 \boldsymbol{g}^T \boldsymbol{H} \boldsymbol{g} \tag{3.7}$$

$\epsilon \boldsymbol{g}^T \boldsymbol{g}$ je očekivano "poboljšanje" tj. približavanje minimumu s obzirom na nagib funkcije, a $g^T \boldsymbol{H} \boldsymbol{g}$ može se promatrati kao korekcija s obzirom na zakrivljenost funkcije [4]. Kada je taj član prevelik, korak gradijentnog spusta povećava iznos funkcije gubitka, umjesto da ga smanjuje. Optimizacijski postupak je prvog stupnja jer se u obzir uzimaju isključivo gradijenti funkcije gubitka. Da je funkcija gubitka aproksimirana razvojem u Taylorov red prvog stupnja, ta bi aproksimacija predvidjela smanjenje funkcije gubitka za iznos $\epsilon \boldsymbol{g}^T \boldsymbol{g}$. Međutim, aproksimacija višeg stupnja od prvog jasno pokazuje da postoje utjecaji viših redova, sve do utjecaja reda l. Ti efekti snažno utječu na novodobivene parametre modela te utječu na izbor koraka učenja ϵ. Nova vrijednost

[5]Točan broj slojeva koji mrežu čini vrlo dubokom nije definiran - mreže od 7 i 15 skrivenih slojeva su obje vrlo duboke.

[6]Nakon unazadne propagacije iste pogreške.

dana je s:

$$\hat{y} = (\omega_1 - \epsilon g_1)(\omega_2 - \epsilon g_2)...(\omega_l - \epsilon g_l) \tag{3.8}$$

Primjer jednog od utjecaja drugog reda koji se javlja u (3.8) je $\epsilon^2 g_1 g_2 \prod_{i=3}^{l} \omega_i$. Ako je $\prod_{i=3}^{l} \omega_i$ malen, utjecaj je zanemariv. No, utjecaj može biti i eksponencijalna funkcija ako su parametri ω_i za $3 \leq i \leq l$ veći od 1. Odabir adekvatnog koraka učenja je zato težak, jer efekti ažuriranja parametara za jedan sloj snažno ovise o ostalim slojevima [4]. Optimizacijski algoritmi drugog reda adresiraju taj problem računanjem ažuriranja koje uzima u obzir interakcije drugog reda, no iz gornjeg primjera vidljivo je da u vrlo dubokim mrežama utjecaji višeg reda mogu biti znatni. Već su i optimizacijski algoritmi drugog reda računski i memorijski iznimno skupi te nerijetko zahtijevaju brojne aproksimacije, što ih čini neupotrebljivima za tipične duboke modele. Izgradnja optimizacijskog algoritma n-tog reda za $n > 2$ stoga se čini besmislenom.

Normalizacija grupe omogućuje elegantan način reparametrizacije gotovo svake duboke neuronske mreže - ulazni podaci/aktivacije se prvo normaliziraju, a zatim linearno preslikavaju koristeći novouvedene parametre sloja. Time se umanjuje problem "koordiniranja ažuriranja" kroz mnogo slojeva neuronske mreže [8]. Normalizacija grupe može se primijeniti na ulaz ili skriveni sloj duboke neuronske mreže. Neka je $\boldsymbol{H}$ mini-grupa aktivacija sloja čije se jedinice normaliziraju, uređena na način da se aktivacije sloja za isti primjer nalaze u retcima matrice. Normalizirana matrica $\boldsymbol{H}$ glasi:

$$\boldsymbol{H}' = \frac{\boldsymbol{H} - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \tag{3.9}$$

$\boldsymbol{\mu}$ je vektor koji sadrži srednje vrijednosti svake jedinice, a vektor $\boldsymbol{\sigma}$ sadrži standardne devijacije svake jedinice. U jednadžbi (3.9), $\boldsymbol{\mu}$ i $\boldsymbol{\sigma}$ se proširuju iz vektora u matrice, kopiranjem redaka dok ne nastanu matrice istih dimenzija kao $\boldsymbol{H}$, a dijeljenje se izvodi po elementima. Unutar retka matrice $\boldsymbol{H}$, svaki element (aktivacija) normalizira se uz odgovarajuće μ_j i σ_j, odnosno element $H_{i,j}$ normalizira se oduzimanjem μ_j i dijeljenjem s σ_j. Otuda dolazi naziv normalizacija *grupe* - $\boldsymbol{\mu}$ i $\boldsymbol{\sigma}$ računaju se po dimenziji mini-grupe. Ostatak neuronske mreže tretira $\boldsymbol{H}'$ na potpuno isti način na koji je bez normalizacije grupe tretirao $\boldsymbol{H}$. Dakle, za vrijeme učenja vektor srednjih vrijednosti dan je s:

$$\boldsymbol{\mu} = \frac{1}{m} \sum_j \boldsymbol{H}_{:,j} \tag{3.10}$$

Analogno, vektor standardnih devijacija:

$$\boldsymbol{\sigma} = \sqrt{\delta + \frac{1}{m} \sum_j (\boldsymbol{H}_{:,j} - \boldsymbol{\mu}_j)^2} \tag{3.11}$$

U izrazu 3.11 $\boldsymbol{\mu}_j$ je vektor-stupac popunjem srednjom vrijednosti μ_j iz originalnog vektora $\boldsymbol{\mu}$. δ je vrlo malena vrijednost (npr. 10^{-8}) koja služi za izbjegavanje dijeljenja s nulom pri unaprijednom i unazadnom prolasku.

Prijašnji pristupi uključivali su tehnike poput proširenja funkcije gubitka kaznom u svrhu postizanja normalizirane statistike aktivacija ili na renormalizaciju statistike jedinice nakon svakog koraka gradijentnog spusta. Prvi pristup uglavnom bi rezultirao nedovoljno dobrom normalizacijom, a drugi velikom količinom izgubljenog vremena, pošto bi algoritam unazadne propagacije pogreške uzastopce predlagao promjenu srednje vrijednosti i varijance kroz gradijente, a normalizacijski korak uzastopce poništavao tu promjenu. S druge strane, normalizacija grupe reparametrizira model na način da neke jedinice po definiciji budu standardizirane - svodi distribuciju njihovih aktivacija na jediničnu Gaussovu razdiobu. Posljedica toga je da gradijent neće predlagati promjenu koja se odnosi samo na srednju vrijednost ili standardnu devijaciju distribucije aktivacija jedinice, budući da je njihov utjecaj dokinut normalizacijom [4].

Za vrijeme zaključivanja, $\boldsymbol{\mu}$ i $\boldsymbol{\sigma}$ mogu se zamijeniti pomičnim prosjecima, koristeći primjerice eksponencijalno otežani pomični prosjek koji se prikuplja za vrijeme trajanja učenja. To modelu omogućuje evaluaciju na jednom primjeru, a da pritom definicije $\boldsymbol{\mu}$ i $\boldsymbol{\sigma}$ ne ovise o mini-grupi.

Povratkom na hipotezu $\hat{y} = x\omega_1\omega_2...\omega_l$, jasno je da se većina poteškoća prisutnih kod učenja takvog modela može rješiti normalizacijom posljednjeg skrivenog sloja h_{l-1}. Neka je x iz jedinične Gaussove razdiobe, tada je i h_{l-1} iz Gaussove razdiobe jer je transformacija od x do h_{l-1} linearna. Međutim, h_{l-1} nije nulte srednje vrijednosti i jedinične varijance poput x. Nakon primjene normalizacije grupe, nastaje normalizirani $\hat{h}_{l-1}$ kojim se jediničnost razdiobe obnavlja. Kakvo god da je ažuriranje parametara nižih slojeva, $\hat{h}_{l-1}$ ostaje iz jedinične Gaussove razdiobe. Izlazni sloj modela $\hat{y}$ tada se može učiti kao jednostavna linearna funkcija $\hat{y} = \omega_l\hat{h}_{l-1}$. Učenje je sada pojednostavljeno jer parametri nižih slojeva dubokog modela imaju slabiji utjecaj - njihov izlaz je uvijek renormaliziran na jediničnu Gaussovu razdiobu. U nekim rubnim slučajevima niži slojevi mogu utjecati - postavljanje težina nižeg sloja na 0 (degeneracija izlaza) ili promjena predznaka jedne ili više težina iz nižeg sloja (moguće zrcaljenje odnosa $\hat{h}_{l-1}$ i y). Takve situacije su rijetke, a bez normalizacije, gotovo svako ažuriranje parametara znatno bi utjecalo na statistiku h_{l-1}. Dakle, normalizacija grupe uzrokuje znatno lakše učenje ovakvog modela, no cijena koja je plaćena jest beskorisnost nižih slojeva modela. Za primjer linearnog modela koji je dan u ovom poglavlju, nakon normalizacije niži slojevi više nemaju nikakav štetan utjecaj, no niti djelotvoran. Normalizacijom su dokinuti momenti prvog i drugog reda (srednja

vrijednost i varijanca) - jedino na što linearna mreža ima utjecaj. S druge strane, u dubokoj neuronskoj mreži s nelinearnim aktivacijskim funkcijama niži slojevi ostaju korisni jer mogu provoditi nelinearne transformacije nad podacima. Normalizacijom grupe standardiziraju se srednja vrijednost i varijanca svake jedinice s ciljem stabilizacije dinamike učenja, no odnosi između jedinica i nelinearna statistika svake pojedine jedinice mogu se mijenjati [4].

Normalizacija srednje vrijednosti i standardne devijacije jedinice može smanjiti ekspresivnu moć neuronske mreže koja sadrži tu jedinicu. Da bi se očuvala ekspresivna moć neuronske mreže, uobičajeno je matricu aktivacija $\boldsymbol{H}$ zamijeniti izrazom:

$$\boldsymbol{a} = \boldsymbol{\gamma}\boldsymbol{H}' + \boldsymbol{\beta} \tag{3.12}$$

Dakle, koristi se (3.12) umjesto samo normalizirane matrice $\boldsymbol{H}'$, gdje su $\boldsymbol{\gamma}$ i $\boldsymbol{\beta}$ parametri koji se uče kroz algoritam unazadne propagacije pogreške i koji omogućuju novoj varijabli da ima proizvoljnu srednju vrijednost i standardnu devijaciju. Postavlja se pitanje zašto nakon normalizacije uvesti parametre koji ponovno dozvoljavaju matrici aktivacija proizvoljne srednje vrijednosti i varijance. Prije svega, uvođenjem (3.12) i dalje se može predstavljati ista familija funkcija nad ulazom kao i prije. Međutim, s (3.12) mijenja se dinamika učenja. Bez (3.12), srednje vrijednosti u $\boldsymbol{H}$ bile su određene složenim interakcijama između parametara slojeva nižih od sloja za kojeg se računa $\boldsymbol{H}$. S druge strane, u (3.12) srednju vrijednost određuje jedino parametar $\boldsymbol{\beta}$. Analogno i za varijancu. S obzirom na (3.9) i (3.12), normalizaciju grupe možemo promatrati kao metodu adaptivne reparametrizacije [4]. Uz novu parametrizaciju, učenje gradijentim spustom je olakšano. Nadalje, s obzirom da većina slojeva neuronskih mreža poprima oblik $\phi(\boldsymbol{X}\boldsymbol{W} + \boldsymbol{b})$, gdje $\phi(\cdot)$ predstavlja proizvoljnu nelinearnu aktivacijsku funkciju (npr. ReLU), postoji prostor za odabir varijable koju se normalizira - ulaz[7] $\boldsymbol{X}$ ili transformacija ulaza $\boldsymbol{X}\boldsymbol{W} + \boldsymbol{b}$. Preporuča se normalizirati transformirane ulaze. Ulaz u sloj neuronske mreže tipično je izlaz prijašnjeg sloja (nelinearna funkcija linearne transformacije) čiji će se oblik vjerojatnosne razdiobe vrlo vjerojatno mijenjati tijekom učenja[8], dok je za transformaciju $\boldsymbol{X}\boldsymbol{W} + \boldsymbol{b}$ izglednije da bude sličnija Gaussovoj razdiobi pa bi normalizacijom dobivena distribucija trebala biti stabilnija [8]. Također, u tom se slučaju $\boldsymbol{X}\boldsymbol{W} + \boldsymbol{b}$ može zamijeniti s $\boldsymbol{X}\boldsymbol{W}$, budući da pomak $\boldsymbol{b}$ postane redundantan uz primjenu $\boldsymbol{\beta}$ u reparametrizaciji (3.12). Zbog svega navedenog, posljedica normalizacije grupe je često brža konvergencija modela, odnosno isti ili bo-

[7]Ulaz se, osim na ulazne podatke modela, odnosi i na izlazne aktivacije prijašnjeg sloja $\phi(X_{l-1}W_{l-1} + b_{l-1})$

[8]Izlazi nelinearnosti $\phi(\cdot)$ koje se tipično koriste (ReLU) ne mogu biti normalno distribuirani.

lji rezultati uz manje optimizacijskih koraka (ušteda vremena) te veću stopu učenja [4].

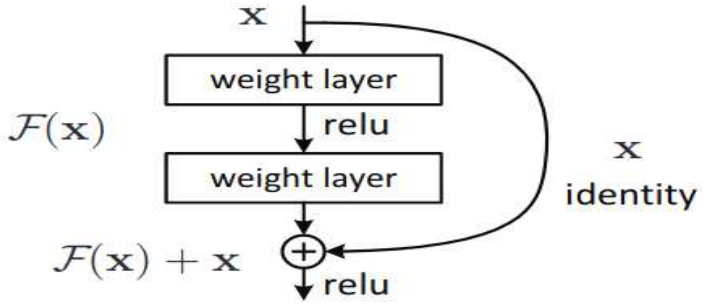
Za slojeve konvolucijske neuronske mreže, potrebno je kod normalizacije imati na umu svojstva konvolucije - da različiti elementi iste mape značajki, na različitim lokacijama, budu normalizirani na isti način. Da bi se to postiglo, zajedno se normaliziraju aktivacije mini-grupe, po svim lokacijama. Za mini-grupu veličine m i mape značajki dimenzija $p \times q$, normalizacija se provodi po efektivnoj mini-grupi veličine mpq. Dakle, svaka mapa značajki (dimenzija kanala c) normalizira se po svim svojim prostornim dimenzijama $p \times q$ zajedno s dimenzijom mini-grupe m. Uči se jedan par parametara γ i β po mapi značajki, umjesto po aktivaciji. Na taj način očuvana je statistika na razini mape značajki, budući da se unutar nje primjenjuju isti μ i σ neovisno o prostornoj lokaciji. Također, tijekom zaključivanja, na sve aktivacije unutar iste mape značajki primjenjuje se ista naučena linearna transformacija [8].

3.3. Duboko rezidualno učenje

Težina učenja duboke neuronske mreže proporcionalna je njenoj dubini. Okvir za rezidualno učenje dubokih neuronskih mreža olakšava učenje "dubljih" neuronskih mreža. Rezidualno učenje odnosi se na pristup u kojem se slojevi neuronske mreže eksplicitno reformuliraju tako da uče *rezidualne funkcije* [6]. Pojednostavljeno, u matematičkom smislu rezidual je devijacija ili pogreška pretpostavljene vrijednosti u odnosu na mjerenu. U kontekstu strojnog učenja, skup točaka $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$; $\boldsymbol{x}_i \in \mathbb{R}^2$ mogli bismo, primjerice opisati modelom linearne regresije: $\hat{y}_i = \boldsymbol{W}^T \boldsymbol{x}_i + \boldsymbol{b}$, $y \in \mathbb{R}$ minimizacijom srednje kvadratne pogreške kao funkcije gubitka. Pogreška ili rezidual predviđanja $\boldsymbol{Y}$ u zavisnosti od $\boldsymbol{X}$ glasila bi: $\hat{\boldsymbol{Y}} - \boldsymbol{Y}$.

Pokazalo se da mnogi zadaci u računalnom vidu profitiraju od povećanja dubine neuralnih modela. Međutim, učenje boljih (dubljih) modela nije takve jednostavnosti da se svodi na slaganje više uzastopnih slojeva. Autori u [6] opisuju to na sljedeći način: "Kada duboke neuronske mreže imaju uvjete za konvergenciju, javlja se problem *degradacije* - s povećanjem dubine neuronske mreže, mjera točnosti ode u zasićenje, a potom rapidno degradira. Neočekivano, do degradacije ne dolazi zbog prenaučenosti i dodavanje novih slojeva u prikladno dubok model vodi do veće pogreške na skupu za učenje." Ako ne poraste samo pogreška na testnom skupu podataka, već i na skupu za učenje, onda navedenu degradaciju ne uzrokuje prenaučenost. Da je u pitanju prenaučenost, pogreška na skupu za učenje bi se smanjila, a na ispitnom skupu povećala u odnosu na model s manje slojeva. Povećanje obje pogreške (s povećanjem

dubine) podsjeća na podnaučenost modela, ali tipična podnaučenost je upravo kada model *nema kapacitet* za opisati stvarnu funkciju koja je generirala podatke - kada bismo pravcem modelirali skup točaka kojeg je generirao polinom trećeg stupnja, a u ovom slučaju je povećanjem dubine kapacitet modela povećan. Informativnije je razmišljati na način da povećanje pogreške na skupu za učenje u ovome slučaju ukazuje na različite težine optimizacijskih problema za različite sustave. Rezidualno učenje olakšava optimizacijski problem vrlo duboke neuronske mreže na sljedeći način: neka se razmatra nešto plića arhitektura i njezin dublji pandan u kojem je nadodano još slojeva. Dublji model se može konstruirati na način da nadodani slojevi budu *jedinična preslikavanja*, a ostali slojevi se "kopiraju" iz naučene pliće inačice modela. Dobiveni model je zapravo jednak plićoj varijanti uz nadodana jedinična preslikavanja, a ideja je da zbog toga dublji model nebi trebao imati veću pogrešku na skupu za učenje od svoje pliće varijante, kao što je objašnjeno u [6]. Slika 3.5 prikazuje rezidualnu konvolucij-



Slika 3.5: Rezidualna konvolucijska jedinica koja se sastoji se od dva konvolucijska sloja. Ulaz prvog sloja dovodi se jediničnim preslikavanjem na izlaz sljedećeg sloja s kojim se zbraja neposredno prije nelinearnosti.

sku jedinicu koja se sastoji od dva sloja čiji se parametri uče te jedinično preslikavanje ulaza prijašnjeg sloja na izlaz sljedećeg sloja. Na izlazima skrivenih slojeva koristi se nelinearna aktivacijska funkcija. Intuicija je sljedeća: umjesto da uzastopni blokovi slojeva izravno uče željeno, nepoznato ulazno-izlazno preslikavanje, slojevima se ekspicitno omogućuje da uče rezidualno preslikavnje. Ako je željeno, nepoznato preslikavanje $\boldsymbol{H}(x)$, nelinearni slojevi uče preslikavanje $\boldsymbol{F}(x) := \boldsymbol{H}(x) - \boldsymbol{x}$. Bez rezidualne veze, blok bi računao $\boldsymbol{F}(\boldsymbol{x})$, a s rezidualnom vezom računa $\boldsymbol{F}(\boldsymbol{x}) + \boldsymbol{x}$. Preciznije, rezidualnu konvolucijsku jedinicu na slici 3.5 opisuje izraz:

$$\boldsymbol{y} = F(\boldsymbol{x}, \{\boldsymbol{W}_i\}) + \boldsymbol{x} \tag{3.13}$$

$\boldsymbol{x}$ i $\boldsymbol{y}$ označavaju ulaz i izlaz bloka. Funkcija $F(\boldsymbol{x}, \{\boldsymbol{W}_i\})$ predstavlja rezidualno preslikavanje koje se uči. Konkretno, za primjer na slici 3.5 s dva sloja unutar rezidualne

jedinice, jednom rezidualnom vezom i nelinearnostima: $F = \boldsymbol{W}_2\sigma(\boldsymbol{W}_1\boldsymbol{x})$, gdje su $\boldsymbol{W}_2$ i $\boldsymbol{W}_1$ parametri slojeva, a σ je ReLU nelinearnost. Dodatno, prema prvom članku o dubokim rezidualnim modelima [6], ako drugi sloj na slici 3.5 nije izlazni sloj modela, uobičajeno je da njegov izlaz bude $\sigma(\boldsymbol{y})$. S druge strane, u novijem članku [7], isti autori predlažu (i potkrepkljuju rezultatima) drugačiji raspored operacija unutar rezidualne jedinice; nelinearnost (ReLU) prije linearnog preslikavanja[9]. Nadalje, da bi se $\boldsymbol{F}$ i $\boldsymbol{x}$ mogli zbrojiti, potrebno je da budu istih dimenzija. Ako to nije slučaj (slojevi su uzrokovali promjenu dimenzija), preskočna veza može, umjesto jedinične, predstavljati općenitu linearnu transformaciju $\boldsymbol{W}_s$ nakon koje će se dimenzije poklapati:

$$\boldsymbol{y} = F(\boldsymbol{x}, \{\boldsymbol{W}_i\}) + \boldsymbol{W}_s\boldsymbol{x} \tag{3.14}$$

Formulacija $\boldsymbol{F}(\boldsymbol{x}) + \boldsymbol{x}$ u unaprijednim neuronskim mrežama realizira se "preskočnim" vezama, kojima se ulaz ranijeg sloja izravno dovodi na izlaz kasnijeg sloja, bez ikakve transformacije između - jedinično preslikavanje. Izlaz kasnijeg sloja se jednostavno zbraja s ulazom koji mu je izravno doveden iz ranijeg sloja preskočnom vezom. Preskočne veze predstavljaju jediničnu matricu u kontekstu transformacije matričnim umnoškom. Posljedično, preskočne veze ne unose dodatne parametre u neuronsku mrežu i ne uzrokuju povećanje računalne složenosti[10]. Neuronska mreža se i dalje u cijelosti može učiti s kraja na kraj. Reprezentacije koje putuju kroz konvolucijsku neuronsku mrežu, tipično će biti dimenzija (N, C, H, W), gdje je N veličina mini-grupe, C broj kanala (broj mapi značajki), a H i W prostorne dimenzije svake mape značajki. Kod korištenja rezidualnih veza u konvolucijskim neuronskim mrežama, umjesto $\boldsymbol{W}_s$ iz izraza (3.14) može se, primjerice, koristiti 1×1 konvoluciju ako je potrebno mijenjati broj kanala od $\boldsymbol{x}$ i poduzorkovanje (npr. bilinearno) za dimenzije H i W. Primjer tipične implementacije je ResNet18[11] kod koje se dimenzije H i W međureprezentacija između blokova mijenjaju kroz parametar pomaka[12], a odgovarajuće dimenzije za operaciju $F(\boldsymbol{x}) + \boldsymbol{x}$ dobivaju se 1×1 konvolucijom.

Smatra se da je u načelu lakše optimirati rezidualno preslikavanje nego izvorno, kod kojeg blok izravno uči $H(\boldsymbol{x})$. U krajnosti, smatra se da kada bi jedinično preslikavanje bilo optimalno, jednostavnije bi bilo optimizacijom natjerati rezidual na nulu nego da blok uzastopnih nelinearnih slojeva nauči jedinično preslikavanje [6]. Eksperimentima je pokazano da su "ekstremno" duboke rezidualne neuronske mreže lakše

[9] Primjerice, za drugi sloj jedinice, umjesto: $\boldsymbol{W}\boldsymbol{x}$ -> BN -> zbroj -> ReLU, predložen je redoslijed: BN -> ReLU -> $\boldsymbol{W}\boldsymbol{x}$ -> zbroj. BN je normalizacija grupe, u tekstu zanemarena zbog jednostavnosti.

[10] Ako se zanemari operacija zbroja $F(\boldsymbol{x}) + \boldsymbol{x}$ koje nema u "klasičnoj" arhitekturi

[11] Konvolucijska neuronska mreža s rezidualnim vezama, sastavljena od 18 slojeva

[12] Konvolucija uz $pomak = 2$, dimenzije filtera 3×3, bez nadopunjavanja

za optimizaciju od klasičnih "ekstremno" dubokih neuronskih mreža kod kojih bi se samo naslagao velik broj uzastopnih slojeva. Točnije, kod klasične inačice pogreška na skupu za učenje raste s povećanjem dubine neuronske mreže, dok duboke rezidualne mreže profitiraju od značajnog povećanja dubine, tj. evaluacijska metrika je bolja. U praksi, nije naročito izgledno da jedinična preslikavanja budu optimalna, no reformulacije opisane izrazima (3.13) i (3.14) pomažu kroz predkondicioniranje problema - kasniji sloj vidi isti skup informacija kao i prijašnji, a u slučaju da je optimalna funkcija dovoljno blizu jediničnom preslikavanju, smatra se da je iz perspektive optimizacije jednostavnije naučiti perturbacije s obzirom na jedinično preslikavanje (kako promijeniti ulazne informacije) nego učiti novu funkciju. Empirijski je pokazano da naučene rezidualne funkcije općenito imaju malene odzive, što ide u prilog teoriji da je jedinično preslikavanje razumno predkondicioniranje [6].

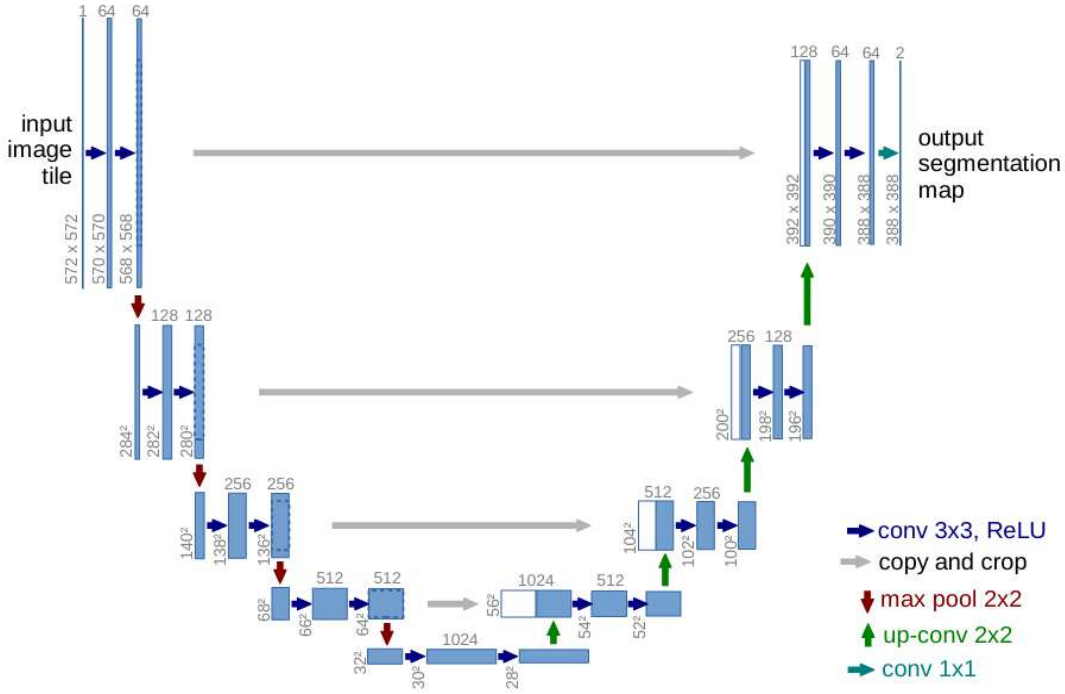
3.4. U-Net arhitektura

Jedna uobičajena primjena konvolucijskih neuronskih mreža odnosi se na klasifikacijske zadatke na razini slike kod kojih je, uz danu sliku na ulazu, izlaz jedna od mogućih K kategorija. Međutim, postoje mnogi zadaci u računalnom vidu, poput monokularne procjene dubine, gdje željeni izlaz uključuje i lokalizaciju, odnosno predikcija se dodjeljuje na razini piksela - *gusta predikcija*. Arhitektura U-Net [15] nadograđuje se na *potpuno konvolucijsku arhitekturu* [13] koja je također osmišljena za zadatke guste predikcije, poput semantičke segmentacije. Potpuno konvolucijska arhitektura zamjenjuje potpuno povezane slojeve konvolucijom, a koristi naduzorkovanje i preskočne veze između slojeva kojima osigurava bogatije međureprezentacije. Pokazuje se da su navedeni arhitekturalni elementi važni u konvolucijskim neuronskim mrežama, posebno za zadatke guste predikcije kod kojih, pošto je predikcija na razini piksela, pitanje nije samo *što*, nego i *gdje*.

Nekorištenje potpuno povezanih slojeva zasniva se na njihovim osnovnim nedostacima - fiksna dimenzionalnost ulaza, neisplativost sa stajališta računalnih resursa te sklonost prenaučenosti. Zbog fiksne dimenzionalnosti, ulaz potpuno povezanog sloja uvijek mora biti iste, odgovarajuće dimenzionalnosti, što stvara dodatnu komplikaciju pri radu modela s ulaznim slikama proizvoljnih dimenzija. To se, doduše, može riješiti unutar same mreže, operacijama poput sažimanja. Veći problem je cijena računalnih resursa, a najveći problem je sklonost prenaučenosti. Ako je na ulazu potpuno povezanog sloja mapa značajki dimenzija $H \times W$, tada svaki od n neurona potpuno povezanog sloja mora imati $HW + 1$ (uz pomak) parametara. Promatrajući potpuno

povezani sloj kroz prizmu konvolucije, na svaki neuron može se gledati kao na operaciju konvolucije s prozorom dimenzija $H \times W$ - svaki neuron "vidi" cijelu mapu značajki na svome ulazu. Dodatno, da bi reprezentacija unutar skrivenog sloja bila dimenzija ulazne mape značajki - što je zgodno za analizu i interpretaciju funkcije koju skriveni sloj obavlja nad ulaznom mapom značajki, potrebno je HW neurona. Tipično, klasifikacijski modeli s kategoričkom distribucijom na izlazu (npr. klasifikacija slike), koriste potpuno povezane slojeve kao izlazne. Potpuno konvolucijski model zamjenjuje izlazni potpuno povezani sloj konvolucijom 1×1, koja je više nego 5 puta brža od izlaznog potpuno povezanog sloja [13]. Nadalje, konvolucija 1×1 prirodan je odabir za zadatke guste predikcije, zbog 2D mapa kakve daje na izlazu. Također, neovisna je (za razliku od potpuno povezanog sloja) o dimenzionalnosti ulaza te čuva izvornu "prostornost" mape značajki.

Na različite međureprezentacije iz različitih slojeva konvolucijske neuronske mreže može se promatrati kao na hijerarhiju značajki (poglavlje 3). Budući da modeli za gustu predikciju osim prepoznavanja i lokaliziraju, pokazalo se da ih je moguće unaprijediti kroz izravnije korištenje lokalnih značajki iz nižih slojeva. Receptivna polja nižih slojeva su manja, a razina detalja lokalnih značajki je finija, veća. Način na koji se to može implementirati je dodavanjem preskočnih veza između odgovarajućih slojeva kodera i dekodera, gdje preskočne veze pritom predstavljaju konkatenaciju mapi značajki, za razliku od matričnog zbroja iz 3.3. Konkatenacija mapi značajki iz "finih" i "grubih" slojeva pomaže modelu da njegove lokalne predikcije poštuju globalnu strukturu. Međutim, kao i kod rezidualnih veza iz 3.3, dimenzije reprezentacija koje se konkateniraju mogu se razlikovati pa ih je neposredno prije konkatenacije potrebno izjednačiti. Konkatenacija se obavlja za svaki primjer iz mini-grupe N po dimenziji kanala C - ostaju prostorne dimenzije $H \times W$. Drugim riječima, ako je tenzor (N, C, H, W) izlaz sloja l, potrebno je njegove "prostorne" dimenzije H i W svesti na $H^{'}$ i $W^{'}$ koje odgovaraju izlaznom tenzoru sloja $l^{'}$ dimenzija $(N, C^{'}, H^{'}, W^{'})$. Postoji i izbor - naduzorkovati nižu rezoluciju ili poduzorkovati višu? Zamisao U-Net arhitekture, prikazane na slici 3.6 je takva da se prirodno nameće naduzorkovanje reprezentacije koja je niže rezolucije. Podaci kroz prikazanu arhitekturu putuju na način da prvo prolaze kroz put "kontrakcije", koji podrazumijeva put s lijeva do sredine (dno slova "u"). Nakon toga, od sredine na desno prolaze put "ekspanzije". Prikazanu arhitekturu može se podijeliti na *koder* i *dekoder*, gdje koder odgovara putu kontrakcije, a dekoder putu ekspanzije - *koder-dekoder arhitektura*. Prema tome, U-Net arhitkura spada u kategoriju koder-dekoder arhitektura. Ulaz u koder je ulaz u model, primjerice RGB slika. Izlaz kodera je konačna skrivena reprezentacija - niskodimenzionalna reprezen-



Slika 3.6: U-Net arhitektura

tacija ulazne RGB slike. Dakle, koder prima ulaz s malim brojem kanala u odnosu na prostorne dimenzije, odnosno ulaz dimenzija $C \times H \times W$ gdje je $C << HiC << W$. Putem kontrakcije (uzastopnom primjenom konvolucije, sažimanja i nelinearne aktivacije), na izlazu kodera nastaje skrivena reprezentacija ulaza dimenzija $C \times H \times W$, ali uz $C >> H$ i $C >> W$. Drugim riječima, prolazeći put kontrakcije, ulazni podaci (RGB slike) koji su visokodimenzionalih prostornih dimenzija i niskodimenzionalne (tro)kanalne dimenzije postaju reprezentacije visokodimenzionalne kanalne dimenzije i niskodimenzionalnih prostornih dimenzija. Ulaz u dekoder je izlaz kodera. Putem ekspanzije, reprezentacije (uzastopnom primjenom konvolucije i naduzorkovanja) prolaze obrnuti put - prostorne dimenzije rastu, a dimenzija kanala se smanjuje, dok u konačnici na izlazu modela ne bude mapa prostornih dimenzija $H \times W$ istovjetnih ulazu modela.

Kontrakcijski put, odnosno koder, slijedi tipičnu arhitekturu konvolucijske neuronske mreže. Konkretno, koder prikazan na slici 3.6 sastoji se od uzastopne primjene dvije konvolucije 3×3 (bez nadopunjavanja - smanjuju prostornu rezoluciju za 2), gdje nakon svake konvolucije slijedi ReLU aktivacija, a nakon obje sažimanje maksimumom 2×2 (uz pomak 2) za poduzorkovanje faktorom 2. Nakon svakog sažimanja maksimumom, sljedeće dvije konvolucije sadrže dvostruko veći broj filtera (udvostručuje se broj mapi značajki) u odnosu na prethodne dvije konvolucije. Na slici su opera-

cija konvolucije i ReLU aktivacije označene plavom strelicom, a sažimanje maksimumom crvenom. Navedene operacije ponavljaju se i u konačnici se od ulaza dimenzija $3 \times 572 \times 572$ dobiva izlaz kodera dimenzija $1024 \times 28 \times 28$. Ekspanzijski put, odnosno dekoder, sastoji se od operacije naduzorkovanja prostornih dimenzija reprezentacije korištenjem tehnike interpolacije (bez parametara za učenje), a naduzorkovati se može i pomoću transponirane konvolucije, kod koje se parametri naduzorkovanja uče jer se radi o konvolucijskim filterima, samo je matrica konvolucije transponirana. U svakom slučaju, nakon (ili u sklopu) naduzorkovanja slijedi prepolavljanje broja mapi značajki, kojim se dobiva simetrija između kodera i dekodera - broj mapi značajki u dekoderu prije konkatenacije jednak je broju mapi značajki iz odgovarajućeg sloja u koderu. Na slici 3.6, to se postiže konvolucijom 2×2 kod koje je broj konvolucijskih filtera dvostruko manji od broja kanala ulazne reprezentacije. Nakon konvolucije 2×2, trenutna reprezentacija u dekoderu se pomoću preskočne veze konkatenira s odgovarajućom (onom s istim brojem kanala) reprezentacijom iz kodera. Na slici 3.6, konkatenacija značajki iz kodera sa značajkama u dekoderu prikazana je bijelim strelicama, a mape značajki nastale konkatenacijom prikazane su kao polubijele, poluplave. Može se dogoditi, što je također slučaj na slici 3.6, da je potrebno odrezati rubove (za mape značajki iz kodera na slici) da bi se rezolucije aktera konkatenacije podudarale. Za konkretnu instancu U-Net arhitekture na slici 3.6, potreba za podrezivanjem nastaje jer se korištenjem konvolucije bez nadopunjavanja gube rubni pikseli - nakon svake konvolucije 3×3 prostorne dimenzije smanje se za 2, što je vidljivo i na slici (rezolucija je zapisana uz donji dio mape značajki). Nakon naduzorkovanja, konvolucije 2×2 i konkatenacije, slijede ponovno parovi konvolucija 3×3 uz ReLU aktivacije kao i u koderu, ali u dekoderu im je uloga smanjenje broja mapi značajki za faktor 2 - obrnuto u odnosu na koder. Konačno, na izlazu dekodera mape se generiraju konvolucijom 1×1. Na slici 3.6 konvolucija 1×1 označena je strelicom tirkizne boje. Za monokularnu estimaciju dubine, konvolucijska neuronska mreža zasnovana na U-Net arhikteturi na izlazu će konvolucijom 1×1 geneirati jednu mapu - mapu dispariteta, rezolucije koja je jednaka rezoluciji ulazne RGB slike (disparitet svakog piksela ulazne slike).

Zaključno, konvolucija obavlja raspoznavanje uzoraka lokalne i globalne razine te manipulira brojem kanala u koderu i dekoderu. Poduzorkovanje se postiže sažimanjem maksimumom, koje unosi i invarijantnost na lokalne translacije. Naduzorkovanjem u dekoderu, rezolucija mapi značajki raste prema rezoluciji ulaza u model. Konkatenacijom, slojevi dekodera raspolažu s više informacija prilikom konstruiranja izlazne mape - imaju pristup globalnim značajkama pristiglim iz kodera koje se dodatno obrađuju

kroz dekoder, a imaju i pristup lokalnim značajkama nastalim u ranijim slojevima kodera s manjim receptivnim poljem, ali finijim, lokalnim detaljima. Konačnu, izlaznu mapu generira konvolucija 1×1. Konvolucijska neuronska mreža za monokularnu procjenu dubine na kojoj se temelje eksperimenti u ovome radu, dodatno je proširena reziudalnim vezama u koderu (3.3) unutar svakog bloka od dvije uzastopne konvolucije 3×3 (konvolucije 1×1 će pritom poslužiti za izjednačavanje po dimenziji kanala). Pritom, svaki blok koristi tehiniku normalizacije grupe (3.2), poduzorkovanje je implementirano konvolucijom uz $pomak = 2$ (umjesto sažimanja maksimumom), a naduzorkovanje se obavlja algoritmima interpolacije. Nadalje, izlazna mapa disparitеta generirat će se na više rezolucija (umjesto samo na konačnoj) - unutar nekoliko različitih slojeva dekodera.

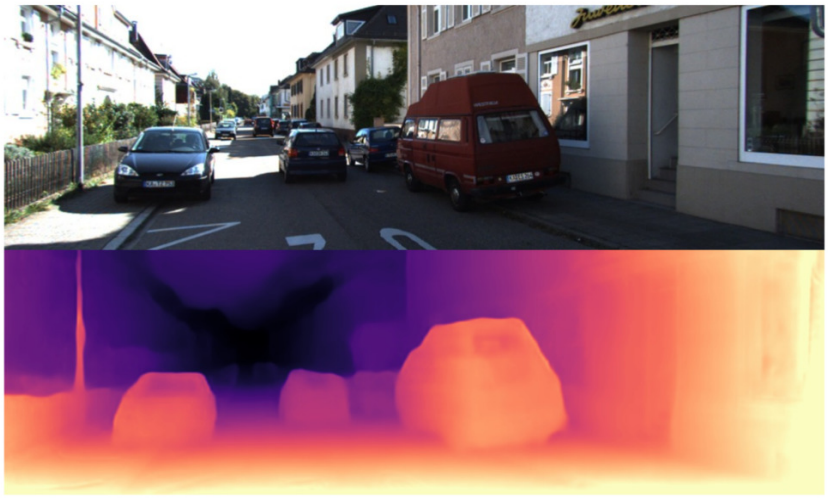
4. Nenadzirano učenje monokularne procjene dubine

Koncept procjene dubine odnosi se na izlučivanje 3D informacija iz scene koristeći 2D informacije dobivene iz kamere. Monokularna rješenja se pritom oslanjaju samo na jednu sliku, odnosno procjenjuju udaljenost između kamere i objekata u sceni na temelju jednog pogleda. Dubina se može i izravno mjeriti senzorima ili procijeniti algoritmima koji se ne temelje na dubokom učenju [14]. Nadalje, dubina se može odrediti *stereo* pristupom, oslanjajući se u svakom trenutku na dva pogleda i *stereo podudaranje*, računajući mape dispariteta pomoću funkcije cijene. Takav pristup je imitacija ljudskog, binokularnog vida. Ključna razlika između monokularne i stereo procjene dubine jest ta što je transformacija između kamera kod stereo slučaja unaprijed dobivena kalibracijom. S obzirom na to da je transformacija između dviju kamera unaprijed poznata, postupak stereo podudaranja posjeduje informaciju o skali (2.2), za razliku od monokularne procjene gdje skala nije poznata i transformaciju između uzastopnih pozicija kamere tek treba procijeniti. Također, budući da su disparitet i udaljenost obrnuto proporcionalne veličine (2.2), za vrlo velike udaljenosti stereo slučaj efektivno degradira u monokularni (disparitet se ne manifestira) pa je u takvim situacijama stereo sustav beskoristan. Nadalje, dubinu je moguće izravno mjeriti korištenjem senzora, poput RGB-D kamera i LIDAR senzora. RGB-D kamerom moguće je dobiti gustu dubinsku mapu, na razini piksela, za RGB sliku. Međutim, domet mjerenja je ograničen, a problem je i osjetljivost na sunčevu svjetlost za primjene u vanjskim, otvorenim okruženjima [17]. S druge strane, LIDAR daje precizna mjerenja dubine, koja se mogu koristiti i kao oznake za nadzirano učenje te evaluaciju modela za monokularnu procjenu dubine (npr. KITTI), ali 3D mape koje generira su rijetke. Osim toga, faktori poput fizičkih dimenzija i potrebe za napajanjem navedenih senzora limitiraju njihovu primjenu u pojedinim područjima, poput dronova u robotici. Uz sve navedeno, procjena guste dubinske mape iz jedne slike motivirana je i niskom cijenom, malim dimenzijama te širokim spektrom mogućih primjena monokularnih

kamera, a duboko učenje pokazuje se kao prikladna tehnika za rješavanje tog zadatka. Ovo poglavlje opisuje jednu od mogućih metoda za nenadzirano učenje monokularne procjene dubine na primjeru *monodepth2*[1] modela za monokularnu procjenu dubine [3]. Na spomenutom modelu temelje se svi eksperimenti napravljeni u sklopu ovog rada. Konkretno, opisan je postupak nenadziranog učenja modela, korištene konvolucijske neuronske mreže za procjenu dubine i vizualnu odometriju te metrike korištene za postupke učenja i evaluacije modela.

4.1. Zadatak modela

Zadatak sustava za monokularnu procjenu dubine je „jednostavan”; uz danu (jednu) RGB sliku na ulazu, sustav treba generirati odgovarajuću dubinsku mapu na izlazu. Slika 4.1 prikazuje ulaze i odgovarajuće izlaze pojedinih komponenti *monodepth2* mo-



Slika 4.1: Slika iz skupa podataka KITTI (gore) i odgovarajuća procjena dubinske mape (dolje). Slika je preuzeta iz [3].

dela na uzorku iz KITTI skupa podataka. Model za monokularnu procjenu dubine *monodepth2* učen je nenadzirano, a postiže relativnu apsolutnu pogrešku na KITTI Eigen podjeli u iznosu od 0.115 na slikovnim okvirima rezolucije 640×192. Procijenjena dubina je veća tamo gdje su tamniji pikseli, a manja gdje su svjetliji. Hipotetski, vrijednosti unutar dubinske mape mogle bi biti iz cijelog skupa $\mathbb{R}$. Monodepth2 procjenjuje mape dispariteta te ih ograničava na realan interval $[0, 1]$. Dubinske mape dobivaju se preslikavanjem procijenjene mape dispariteta na interval $[D_{min}, D_{max}]$, gdje su D_{min}

[1] https://github.com/nianticlabs/monodepth2

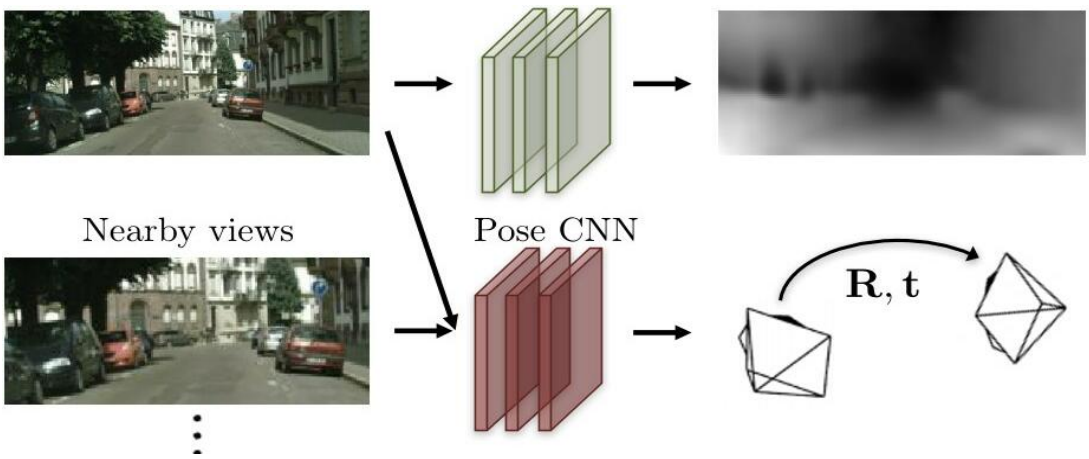
i D_{max} hiperparametri modela koji označavaju minimalnu i maksimalnu dubinu koju model može procijeniti za piksel. Na slici 4.1 model je znatno pogriješio u procjeni područja "dubinske rupe" (područje dubinske mape obojano u crno) kojem je dodijelio maksimalnu dubinu D_{max}. U dubinskoj mapi, ljubičasti dio iznad dubinske rupe odgovara nebu na ulaznoj slici, a nebo je sasvim sigurno udaljenije od kamere nego drveće i dijelovi kuća za koje je model procijenio D_{max}. Prema tome, dubinska mapa bila bi točnija da je kompletno nebo obojano u crno, a što se tiče dijela kojeg model jest obojao u crno na gornjoj slici, teško je odokativno procijeniti nalazi li se unutar ili izvan D_{max} (npr. za $D_{max} = 80$). Za pouzdanu ocjenu takve predikcije potrebna je usporedba s mjerenjem senzora, npr. LIDAR s odgovarajućim dometom. Što se tiče pristupa učenju, nadzirani pristup je izravan jer su podaci označeni; za svaku sliku iz skupa za učenje/validaciju/testiranje postoji odgovarajuća dubinska mapa gdje je označena dubina svakog piksela, kao na slici 4.1. Međutim, nije jednostavno prikupiti takve podatke - visoke kvalitete oznaka te iz raznih okolina u svijetu (važno zbog robusnosti modela). Zbog navedenih razloga, prirodno se nameće nenadzirani pristup - oznake se tijekom učenja ne koriste, neovisno o tome postoje li ili ne. Rezultati modela koji je učen nenadzirano uvelike ovise o smislenosti i primjerenosti osmišljenog postupka učenja, kao što rezultati nadziranog učenja ovise o kvaliteti oznaka. Oba pristupa, naravno, ovise o kvaliteti i specifičnostima samih podataka. Dakle, zadatak modela je da, jednom kada je naučen, za jednu sliku na ulazu generira pouzdanu dubinsku mapu na izlazu. S obzirom na to da taj zahtjev vrijedi tek kada je postupak učenja završen i model „ide u produkciju”, to ništa ne govori o tome kako ga naučiti – po tom pitanju model nije ograničen. Drugim riječima, model ne mora biti učen na način da u svakom koraku učenja na ulaz prima isključivo jednu sliku, za koju na izlazu daje dubinsku mapu, već može primati više slika iste scene (različite poglede na scenu) što je upravo slučaj kod nenadziranog pristupa.

4.2. Postupak nenadziranog učenja na monokularnim podacima

Kada bi model učio nadzirano, mogao bi se, primjerice, sastojati od jedne konvolucijske neuronske mreže koja bi na ulaz primila mini-grupu RGB slika, na izlazu bi geneirala odgovarajuće dubinske mape, zatim bi se izmjerio gubitak u odnosu na "istinite" dubinske mape, a ostatak se svodi na propagaciju gradijenata algoritmom unazadne propagacije pogreške i ažuriranje parametara mreže određenim optimizacijskim

algoritmom. S obzirom da nenadzirani postupak učenja ne koristi oznake, model si samostalno mora generirati signal za učenje. Etablirani postupak za nenadzirano učenje monokularne procjene dubine svodi se na zadatak *rekonstrukcije slike* ili *sinteze pogleda*. Model za monokularnu procjenu dubine može se nenadzirano učiti nad monokularnim nizom slikovnih okvira (M), stereo parovima (S) ili koristeći oboje (M+S). U slučaju monokularnog niza slika, ideja je da se pri svakom koraku učenja modela koristi niz od n uzastopnih slika. Primjerice, za učenje modela monodepth2, zadana konfiguracija koristi $n = 3$, odnosno 3 uzastopne slike $\{I_{t-1}, I_t, I_{t+1}\}$. Slika I_t predstavlja ciljnu sliku koja se rekonstruira na temelju preostalih, izvornih slika $\{I_{t-1}, I_{t+1}\}$. Za rekonstrukciju je potrebno pronaći korespondencije, a da bi to bilo moguće, modelu je potrebna informacija dubine. Prema tome, dubina indirektno doprinosi krajnjem cilju - omogućuje projekciju ciljne slike iz 2D u 3D prostor.

Kod stereo parova (S), za svaki korak učenja koriste se lijeva I_l i desna I_d slika iste scene, a obje mogu imati ulogu ciljne i izvorne slike. Kod kombinacije (M+S), pri svakom koraku učenja koriste se uzastopne slike (M) te stereo par ciljne slike (S).

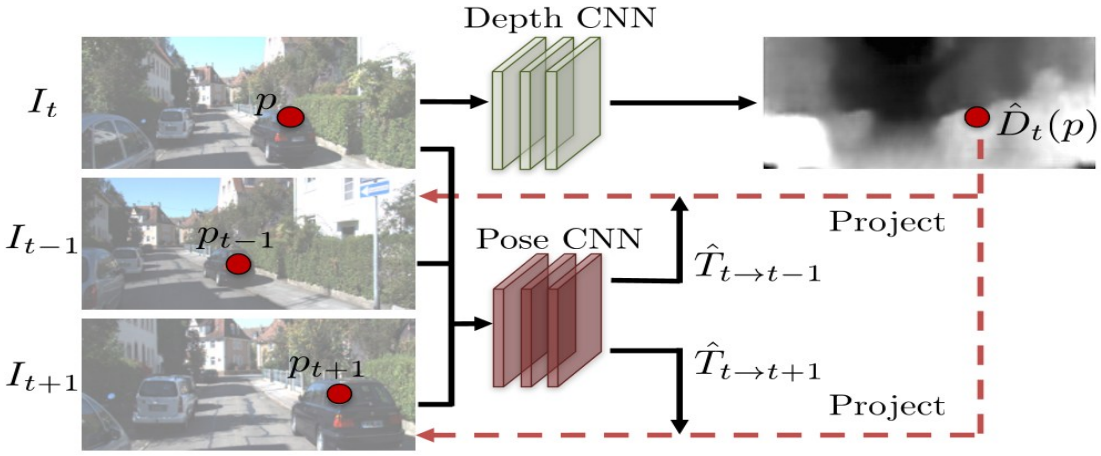


Slika 4.2: Pojednostavljeni prikaz modela za monokularnu procjenu dubine. Jedan primjer za učenje čini niz uzastopnih slika $\{I_{t-1}, I_t, I_{t+1}\}$. Modul za monokularnu procjenu dubine (bijela) na ulaz prima I_t, a na izlazu daje dubinsku mapu D_t. Modul za procjenu relativne poze kamera (crvena) na ulaz prima parove $\{I_{t-1}, I_t\}$ i $\{I_t, I_{t+1}\}$ te za svaki par generira translacijski vektor t i 3 Eulerova kuta iz kojih se gradi rotacijska matrica R. Slika je preuzeta iz [20].

Slika 4.2 prikazuje jednu moguću izvedbu modela za monokularnu procjenu dubine kojeg je moguće učiti nenadzirano. Ulaz u model je niz od nekoliko uzastopnih slika (npr. iz monokularnog videa) koji ujedno tvori jedan primjer za učenje. U „središtu" modela nalaze se dvije konvolucijske neuronske mreže (3.1) – jedna za monoku-

larnu procjenu dubine (bijela) te jedna za procjenu relativne poze kamera (crvena). Uz standardne operacije prisutne u konvolucijskim neuronskim mrežama (3.1.1, 3.1.4), navedene mreže koriste normalizaciju grupe (3.2) i rezidualne veze (3.3). Također, obje mreže su po dizajnu temeljene na U-Net arhitekturi (3.4) i sastoje se od kodera i dekodera.

Niz uzastopnih slika u kameri nastaje uz malen vremenski pomak, stoga je za očekivati da se nekoliko uzastopnih slika (barem dvije) znatno preklapaju po sadržaju, tj. da postoji znatan broj korespondencija - u suprotnom, zadatak rekonstrukcije ciljne slike iz okolnih nebi imao smisla. Na slici 4.2, lijeva gornja slika je ciljna, a ostale su izvorne, okolne slike. Ciljna slika dovodi se na ulaz modula za monokularnu procjenu dubine. Izlaz tog modula je gusta dubinska mapa koja sadrži vrijednosti dubine za svaki piksel ciljne slike. Nadalje, uzastopni niz od n slika dijeli se na parove $\{I_t, I_s\}$, gdje je I_s jedna izvorna slika. Parovi se zatim dovode na ulaz modula za procjenu relativne poze kamera. Ako je niz uzastopnih slika dobiven iz iste kamere, radi se o relativnoj pozi za istu kameru na različitim pozicijama. Model za procjenu releativne poze kamera rješava zadatak vizualne odometrije, koji je nužan da bi rekonstrukcija slike bila moguća. Upravo se u modelu za procjenu relativne poze nalazi jedna od temeljnih razlika između učenja modela na monokularnim (M) i stereo (S) podacima jer je za potonje dovoljno samo jednom, „offline" kalibrirati kamere pa je transformacija između kamera poznata tijekom učenja. Model za procjenu relativne poze kamera tada nije potreban - osnovica b je (pretpostavljamo) fiksna. Monokularni model jedino može procjenjivati disparitet, "zamisliti" da postoji virtualna desna (ili lijeva) kamera na čiju se slikovnu ravninu projiciraju korespondentni pikseli. Kod izračuna dubine na temelju procijenjenog dispariteta iz jednog pogleda, model se može osloniti na intrinsičnu matricu kamere (ako je poznata) te na činjenicu da su disparitet i dubina obrnuto proporcionalni. Alternativno, ako bismo izlaz modula za procjenu dubine tretirali kao $\frac{b}{d}$ (2.20), nenadzirano učenje rekonstrukcijom pogleda ne bi bilo moguće jer je za taj zadatak potrebna dubina kao izdvojena veličina. Slika 4.3 prikazuje tok informacija kod nenadziranog učenja modela za monokularnu procjenu dubine. Slično kao na slici 4.2, $\{I_{t-1}, I_t, I_{t+1}\}$ predstavljaju jedan primjer za učenje - niz uzastopnih slikovnih okvira. I_t je ciljna slika koju se rekonstruira iz I_{t-1} i I_{t+1}, gdje $t-1$ i $t-1$ označavaju jednu sliku "ranije", tj. jednu sliku "kasnije" u odnosu na trenutak t. Koliko vremenski ranije ili kasnije, ovisi o broju slikovnih okvira po sekundi unutar kamere. Također, količina korespondentnih piksela između parova $\{I_t, I_{t-1}\}$ i $\{I_t, I_{t+1}\}$ znatno ovisi o odnosu brzine kretanja kamere kojom je prikupljen skup za učenje i broju slikovnih okvira po sekundi. Ako bi se kamera tijekom prikupljanja skupa za učenje kretala pre-



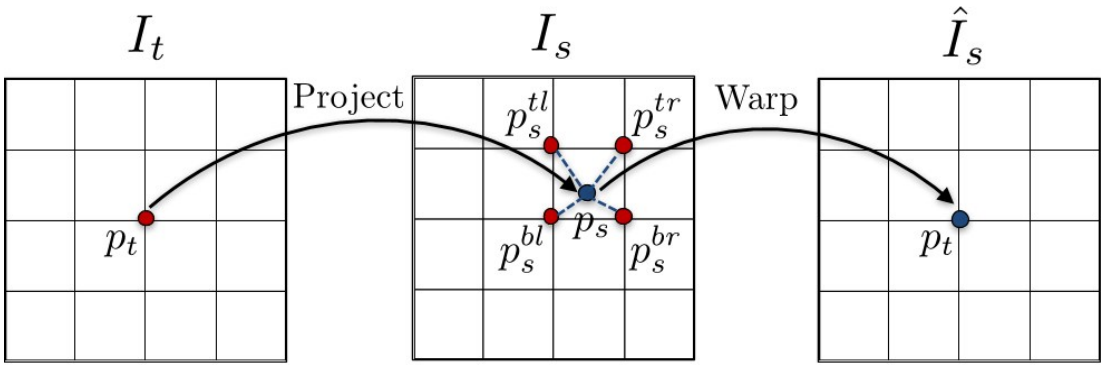
Slika 4.3: Model za monokularnu procjenu dubine uz smjerove projekcija. Pikseli $\{p, p_{t-1}, p_{t+1}\}$ korespondiraju. Piksel p ciljne slike projicira se u 3D prostor na temelju procijenjene dubine i intrinsične matrice kamere. Modul za procjenu relativne poze kamera generira transformacije $\{\hat{T}_{t\to t-1}, \hat{T}_{t\to t+1}\}$, koje omogućuju pronalazak korespondentnih piksela $\{p_{t-1}, p_{t+1}\}$. Ciljna slika rekonstruira se iz izračunatih korespondencija. Slika je preuzeta iz [20].

brzo u odnosu na frekvenciju kojom nastaju slike, količina korespondentnih piksela znatno bi se smanjila. To bi, naime, bilo nepoželjno jer kvaliteta rekonstrukcije ciljne slike ovisi o pronalasku korespondencija između ciljne I_t i okolnih I_{t-1}, I_{t+1} slika. Na slici 4.3 crvenim točkama označeni su korespondentni pikseli $\{p, p_{t-1}, p_{t+1}\}$. Rekonstrukcija slike provodi se na razini piksela - rezolucija dubinske mape i odgovarajuće ulazne slike je jednaka. Kao što je vidljivo na slici, konvolucijska neuronska mreža za procjenu dubine (bijela) na ulaz prima jedino I_t, a na izlazu daje procjenu odgovarajuće dubinske mape $\hat{D}_t(p)$. S druge strane, konvolucijska neuronska mreža za procjenu relativne poze između kamera (pogleda), na ulaz u odvojenim koracima prima parove $\{I_t, I_{t-1}\}$ te $\{I_t, I_{t+1}\}$. Taj model generira procjenu odgovarajuće transformacije koordinatnog sustava za navedene parove - $\hat{T}_{t\to t-1}$ i $\hat{T}_{t\to t+1}$. Nakon toga, rekonstrukciju I'_t slike I_t opisuje izraz:

$$p_s \sim K\hat{T}_{t\to s}\hat{D}_t(p_t)K^{-1}p_t \tag{4.1}$$

U gornjoj formuli, p_t je piksel ciljne slike, a p_s piksel izvorne slike kojeg model smatra korespondencijom za p_t. $\hat{D}_t(p_t)$ je procijenjena dubina za piksel p_t. $\hat{T}_{t\to s}$ je procijenjena transformacija koordinatnog sustava ciljne slike u koordinatni sustav izvorne slike (rotacija i translacija). K je intrinsična matrica kamere, a K^{-1} njezin inverz. Logika izraza (4.1) je sljedeća; p_t je nastao projekcijom 3D točke (2.1.2) na slikovnu ravninu. Projekcija množenjem s K^{-1} i $\hat{D}_t(p_t)$ daje 3D točku definiranu u koordinat-

nom sustavu kamere u trenutku t. Zatim, množenjem s $\hat{T}_{t\to s}$ (2.1.6), definicija točke p_t prelazi u koordinatni sustav druge kamere[2]. Korespondencija piksela p_t je onda njegova projekcija iz 3D koordinatnog sustava druge kamere na njenu slikovnu ravninu - piksel p_s. U svim koordinatnim sustavima (prostor piksela, 3D koordinatni sustavi kamera) točke su zapisane u homogenim koordinatama (2.1.3) kao 4D homogeni vektori. Matrica transformacije $\hat{T}_{t\to s}$, intrinsična matrica K i njezin inverz K^{-1} su dimenzija 4×4. Međutim, koordinate p_s dobivene na ovaj način ne moraju ispasti cjelobrojne. Pikseli se indeksiraju cjelobrojnim koordinatama, stoga je potrebno zaokruživanje ili korištenje neke od metoda uzorkovanja. *monodepth2* koristi bilinearno uzorkovanje [16]. Slika 4.4 prikazuje bilinearno uzorkovanje. Piksel p_t ciljne slike I_t se primjenom



Slika 4.4: Bilinearno uzorkovanje

izraza (4.1) projicira u piksel p_s izvorne slike I_s, ali s realnim koordinatama. Budući da realne koordinate ne indeksiraju jednoznačno piksel, za potrebe rekonstrukcije intenzitet piksela p_s izračunat je bilinearnim uzorkovanjem. Bilinearno uzorkovanje je linearna interpolacija na temelju četiri susjedna piksela (gornji lijevi, gornji desni, donji lijevi, donji desni) od p_s:

$$\hat{I}_s(p_t) = I_s(p_s) = \sum_{n\in\{t,b\},m\in\{l,r\}} \omega_{nm} I_s(p_s^{n,m}) \tag{4.2}$$

$p_s^{n,m}$ označava koordinate susjednog piksela uzorkovane težinom ω_{nm} koja je linearno proporcionalna prostornoj blizini susjednog piksela u odnosu na piksel p_s. Iz tog razloga vrijedi:

$$\sum_{n,m} \omega_{nm} = 1 \tag{4.3}$$

Prema implementaciji [9] koju koristi model monodepth2 za bilinearno uzorkovanje,

[2] Ako je skup za učenje prikupljen jednom kamerom, onda se radi o istoj kameri u različitim vremenskim trenucima

težine ω_{nm} računaju se na sljedeći način:

$$\omega_{nm} = \max(0, 1 - |x_i^s - m|) \max(0, 1 - |y_i^s - n|) \tag{4.4}$$

U gornjem izrazu, (x_i^s, y_i^s) su koordinate korespondentnog piksela p_s iz (4.1). Time je iz izvornog pogleda I_s uzorkovana vrijednost $I_s(p_s)$ korespondentnog piksela, koja ujedno predstavlja vrijednost tog istog piksela rekonstruiranog u ciljnom pogledu $\hat{I}_s(p_t)$. Dakle, piksel p_t u rekonstruiranom pogledu "obojan" je vrijednošću korespondentnog piksela iz izvornog pogleda koji je dobiven transformacijom (4.1) te uzorkovan pomoću (4.2).

4.3. Komponente funkcije gubitka

4.3.1. Fotometrijski gubitak

Jednom kada je ciljni pogled u potpunosti sintetiziran, odnosno rekonstruiran iz izvornog pogleda, moguće je generirati signal za učenje modela kao funkciju rekonstruirane i originalne ciljne slike:

$$L_p = \sum_s \sum_p |I_t(p) - \hat{I}_s(p)| \tag{4.5}$$

Izraz 4.5 predstavlja *gubitak rekonstrukcije* ili *fotometrijski gubitak*, temeljen na L_1 normi originalne i rekonstruirane ciljne slike, gdje se kvaliteta rekonstrukcije interpretira na temelju intenziteta, na razini piksela. Neka je $\{I_1, ...I_n\}$ niz slikovnih okvira koji čine jedan primjer za učenje, gdje je jedan od slikovnih okvira ciljni pogled I_t, a preostali su izvorni pogledi I_s, uz $1 \leq s \leq N, s \neq t$. U izrazu (4.5) s indeksira izvorni pogled, p indeksira koordinate piksela, $I_t(p)$ je intenzitet piksela u originalnom ciljnom pogledu kojeg se rekonstruira, a $\hat{I}_s(p)$ je izvorni pogled projiciran u koordinatni sustav ciljnog pogleda, odnosno rekonstrukcija ciljnog pogleda iz izvornog dobivena na temelju (4.1) i (4.2).

Budući da je signal za učenje L_1 norma rekonstruirane slike u odnosu na ciljnu na razini piksela (fotometrijski gubitak), model se izravno optimizira za taj zadatak, a procjena dubine sudjeluje u postupku učenja kroz izraz za transformaciju iz ciljnog koordinatnog sustava u izvorni (4.1). Međutim, željeni rezultat je model za monokularnu procjenu dubine. Učenje s ovakvim postavkama, arhitekturom i ciljem uzrokuje i uspješno učenje modela za monokularnu procjenu dubine, iako procjena dubina sudjeluje indirektno. Pokazuje se, da bi kvaliteta geometrijskog sustava za sintezu pogleda bila konzistentna, potrebno je da njegovo poimanje geometrije scene te relativnih poza

između kamera što bolje odgovara stvarnosti. Sustav je „natjeran" da kroz formulaciju cilja kao što je sinteza pogleda nauči o među-zadacima poput procjene dubine i relativne poze, koji mu pomažu da izgradi konzistentno vizualno razumijevanje svijeta.

4.3.2. Gubitak glatkosti

Da bi se učenjem dobio bolji model za monokularnu procjenu dubine, pokazuje se da je rekonstrukcijiski gubitak 4.5 pametno proširiti izrazom za zaglađivanje dubinske mape, odnosno dodavanjem *gubitka glatkosti*:

$$L_s = |\partial_x \hat{d}_t| e^{-|\partial_x I_t|} + |\partial_y \hat{d}_t| e^{-|\partial_y I_t|} \tag{4.6}$$

Simbol $\hat{d}_t$ označava mapu dispariteta dobivenu za ciljnu sliku I_t, normaliziranu dijeljenjem svakog elementa mape prosječnom vrijednošću mape. Trik s normalizacijom služi za obeshrabrivanje modela u nastojanju da generalno smanjuje vrijednosti procijenjene dubine da bi smanjio gubitak koji kažnjava model za glatkost (4.6). Ideja je natjerati model da dubine budu glatke, odnosno penalizirati nagle skokove u procjeni vrijednosti dubine. Parcijalne derivacije po obje osi označavaju da (4.6) ovisi o apsolutnoj vrijednosti razlika susjednih dubina po obje osi. Istovremeno, u stvarnosti se na rubovima objekata mogu pojaviti nagli skokovi dubine pa je modelu potrebno dati do znanja da u tom slučaju nije riječ o pogrešnoj procjeni dubine. Rubovi objekata najčešće dovode i do različitih boja susjednih piksela pa stoga kaznu (4.6) eksponencijalno prigušujemo s obzirom na gradijent slike $\partial_x I_t$ i $\partial_y I_t$.

4.3.3. Mjera indeksa strukturalne sličnosti

Pokazuje se da je gubitak moguće dodatno unaprijediti ako se fotometrijski gubitak komplementira *gubitkom strukturalne sličnosti*:

$$SSIM(\boldsymbol{a}, \boldsymbol{b}) = \frac{(2\mu_{\boldsymbol{a}}\mu_{\boldsymbol{b}})(\sigma_{\boldsymbol{ab}} + \epsilon)}{(\mu_{\boldsymbol{a}}^2 + \mu_{\boldsymbol{b}}^2)(\sigma_{\boldsymbol{a}}^2 + \sigma_{\boldsymbol{b}}^2 + \epsilon)} \tag{4.7}$$

Uz dane slike (ili dijelove slika) $\boldsymbol{a}$ i $\boldsymbol{b}$, mjera indeksa strukturalne sličnosti na izlazu daje vrijednost $SSIM(\boldsymbol{a}, \boldsymbol{b}) \in [0, 1]$. Prema tome, najveća sturkturalna sličnost između dvije slike je 1, a minimalna 0. ϵ je malena konstanta koja služi za izbjegavanje scenarija dijeljenja nulom. $\mu_{\boldsymbol{a}}$ je srednja vrijednost slike (ili dijela slike) $\boldsymbol{a}$:

$$\mu_{\boldsymbol{a}} = \frac{1}{n} \sum_{i=1}^{n} a_i \tag{4.8}$$

Analgono vrijedi za $\mu_{\boldsymbol{b}}$. $\sigma_{\boldsymbol{a}}^2$ označava varijancu slike (ili dijela slike) a:

$$\sigma_{\boldsymbol{a}} = \frac{1}{n-1}\sum_{i=1}^{n}(a_i - \mu_{\boldsymbol{a}}^2) \tag{4.9}$$

Analgono vrijedi za $\sigma_{\boldsymbol{b}}^2$. $\sigma_{\boldsymbol{ab}}$ ukazuje na koreliranost dvaju ulaza:

$$\sigma_{\boldsymbol{ab}} = \frac{1}{n-1}\sum_{i=1}^{n}(a_i - \mu_{\boldsymbol{a}})(b_i - \mu_{\boldsymbol{b}}) \tag{4.10}$$

4.3.4. Konačna funkcija gubitka

Funkcija gubitka koju koristi model *monodepth2*, objedinjuje fotometrijski gubitak (4.5) i mjeru strukturalne sličnosti (4.7) u gubitak fotometrijske rekonstrukcije pe:

$$pe(I_t, I_{t'\to t}) = \frac{\alpha}{2}(1 - SSIM(I_t, I_{t'\to t})) + (1-\alpha)||I_t - I_{t'\to t}||_1 \tag{4.11}$$

U izrazu (4.11), hiperparametar α kontrolira težine pojedinih utjecaja (4.7 i 4.5) u gubitku fotometrijske rekonstrukcije. Budući da je $SSIM$ mjera strukturalne *sličnosti*, uz $SSIM(x, y) \in [0, 1]$, prvi pribrojnik predstavlja komplement mjere strukturalne sličnosti, odnosno može se interpretirati kao mjera strukturalne različitosti otežana djelovanjem parametra α. Drugi pribrojnik je izraz (4.5), također otežan djelovanjem parametra α. Nadalje, izraz (4.11) računa se za svaki par $\{I_t, I_{t'\to t}\}$, za svaki piksel, na razini mini-grupe. Postavlja se pitanje kako objediniti gubitke izračunate za različite parove $(I_t, I_{t'\to t})$, a za iste piksele ciljne slike I_t. Jedno rješenje je računati prosjek različitih gubitaka za isti piksel, no *monodepth2* uvodi minimum koji se pokazuje boljim:

$$L_p = \min_{t'} pe(I_t, I_{t'\to t}) \tag{4.12}$$

Dakle, kada se gubitak za isti piksel p_t ciljne slike I_t izračuna kroz sve parove $(I_t, I_{t'\to t})$, konačan gubitak koji odgovara pikselu p_t je minimum po svim parovima. Budući da se unutar svakog primjera mini-grupe nalazi jedna ciljna slika te jedna ili više izvornih, u (4.12) označen je minimum po t'. Minimum je, u ovome slučaju, pametnija mjera od prosjeka - ako je prisutno zaklanjanje u nekim parovima, minimum se pobrine da je gubitak određen isključivo onim parom na kojem nema zaklanjanja (bolja rekonstrukcija). S prosjekom bi i parovi na kojima je prisutno zaklanjanje utjecali na gubitak, odnosno kažnjavali bismo model što ne rekonstruira uspješno one dijelove ciljne slike koji su zaklonjeni na izvornim slikama. Osim zaklanjanja, rubni pikseli slike I_t "izađu" iz vidnog polja zbog gibanja kamere i nema ih na slici I_{t+1} - za rubne piksele I_t ne postoje korespondencije na I_{t+1}. S druge strane, model može na rubovima slike

I_{t-1} tražiti korespondencije kojih na slici I_t nema (ako se kamera giba prema naprijed). Zbog svega navedenog, izraz (4.12) je prikladan.

Konačan oblik funkcije gubitka je kombinacija gubitka fotometrijske rekonstrukcije (4.12) i regularizacijskog gubitka (4.6):

$$L = \mu L_p + \lambda L_s \tag{4.13}$$

Poput α u (4.11), parametri λ i μ u (4.13) kontroliraju težinu utjecaja pojedinih komponenti ukupne funkcije gubitka. Ukupni gubitak (4.13) uprosječuje se po dimenzijama piksela i mini-grupe. Također, uprosječuje se i za različita mjerila dubinske mape. Točnije, modul za procjenu dubine modela monodepth2 generira mape dispariteta na više rezolucija, odnosno u više slojeva dekodera tog modula, umjesto samo u izlaznom. Koriste se 4 mjerila. Ako je ulazna slika rezolucije $H_0 \times W_0$, rezolucije dubinske mape nad kojima se provodi učenje modela glase:

$$W_i = \frac{W_0}{2^i}$$

$$H_i = \frac{H_0}{2^i} \tag{4.14}$$

Dakle, dubinska mapa $\hat{D}_t$ ulazi u funkciju gubitka na 4 rezolucijska mjerila (4.14), za $i \in \{0, 1, 2, 3\}$. Za svako rezolucijsko mjerilo računa se isti gubitak (4.13). Prije izračuna gubitka, svaka dubinska mapa se bilinearnom interpolacijom naduzorkoje[3] na rezoluciju $H_0 \times W_0$. Pokazuje se da računanje gubitka na više rezolucijskih mjerila doprinosi učenju boljeg modela, a intuicija je da se model na taj način usmjerava da sve dubinske mape izgrađuje s obzirom na konačan cilj - što točnija rekonstrukcija ulaza visoke rezolucije [3]. Broj rezolucijskih mjerila je hiperparametar modela.

4.3.5. Maskiranje stacionarnih piksela

U postupak nenadziranog učenja koji je opisan u ovom poglavlju, inherentno je ugrađena pretpostavka o kameri koja se giba i sceni koja miruje. Ta pretpostavka, naravno, nije uvijek točna. Primjerice, ako kamera miruje i nema objekata koji se gibaju - uzastopne slike su identične. Nadalje, gibanje objekata u sceni također modelu stvara probleme jer konvolucijska neuronska mreža za procjenu relativne poze procjenjuje transformacije rotacije i translacije na razini slike. Drugim riječima, transformacije su definirane na razini koordinatnog sustava i sve točke jedne slike podliježu istoj transformaciji. Dakle, model po dizajnu ne uzima u obzir moguće postojanje piksela koji

[3]Osim dubinske mape koju daje izlazni sloj modula, ona nastaje na rezoluciji $H_0 \times W_0$.

pripadaju objektima koji se gibaju - objektima za čiji disparitet nije zaslužno isključivo gibanje kamere, već i njihovo vlastito gibanje. Navedeni problemi mogu se manifestirati kao "dubinske rupe" (regije maksimalne dubine) u dubinskim mapama za vrijeme zaključivanja, za objekte koji se obično gibaju na slikama skupa za učenje. *monodepth2* adresira probleme statične kamere i objekata čije je gibanje samo translatirano gibanje kamere. Pikseli koji se ne mijenjaju kroz nizove slikovnih okvira nerijetko su implikacija statične kamere ili takvih objekata. Navedeni pikseli maskiranjem se isključuju iz funkcije gubitka jer prkose inherentnoj pretpostavci modela da se kamera giba, a svijet miruje. *monodepth2* koristi tvrdu, binarnu masku, $\mu \in \{0, 1\}$ koja se računa tijekom unaprijednog prolaska:

$$\mu = [\![\min_{t'} pe(I_t, I_{t' \to t}) < \min_{t'} pe(I_t, I_{t'})]\!] \tag{4.15}$$

U izrazu (4.15), $[\![]\!]$ su Iversonove zagrade, tj. $\mu = 1$ ako je izraz unutar zagrada istinit, inače 0 (piksel se isključuje). Računanje maske (4.15), odnosno piksela koji se isključuju iz funkcije gubitka, ne unosi nove parametre u model. Ideja je da funkcija gubitka razmatra samo one piksele za koje vrijedi da je rekonstrukcijska pogreška manja između ciljne I_t i rekonstruirane slike $I_{t' \to t}$, nego između ciljne I_t i izvorne $I_{t'}$. Ako to vrijedi, onda se očito ciljna i izvorna slika razlikuju - kamera se giba (ili miruje, a scena se giba). Dakle, cilj je izrazom (4.15) isključiti piksele koji se ne mijenjaju kroz slikovne okvire jer takvi nisu za model nisu informativni, a potencijalno uzrokuju da mreža za procjenu dubine nauči maksimalne dubine za takve regije. Doduše, procjena maksimalne dubine za regije koje se ne mijenjaju kroz niz slika je logična posljedica - ako se ne mijenjaju, disparitet je minimalan, tj. dubina treba biti maksimalna. Primjerice, to je u redu kada je u pitanju nebo koje se vidi kamerom na zemljinoj površini, no za objekte koji su blizu kamere - to je pogrešna procjena. Međutim, izraz (4.15) i dalje ne rješava problem objekata koji se gibaju, uz kameru koja se giba. Za takve piksele, vrijednost maske nije predvidiva. Razlog je jednostavan; za piksel koji pripada objektu koji se giba (uz gibanje kamere) postojat će razlika između ciljne i izvorne slike[4]. Nadalje, transformacija relativne poze odgovorna je za gibanje kamere, no za gibanje samih objekata "nitko nije odgovoran". Neka je ta transformacija uistinu procijenjena na temelju piksela čije je gibanje uzrokovano isključivo gibanjem kamere. Tada, pikseli koji se uz gibanje kamere gibaju i sami, bit će transformirani isključivo s obzirom na gibanje kamere, bez uzimanja njihovog gibanja u obzir. Prema tome, nije predvidivo gdje će se u rekonstruiranoj slici nalaziti takvi pikseli niti koja će strana nejednakosti

[4]Pretpostavlja se da se i piksel i kamera gibaju, na način da im se gibanja međusobno ne poništavaju.

izraza (4.15) prevladati jer nije jasno hoće li pogreška, uz navedene pretpostavke, biti veća prije ili nakon rekonstrukcije.

4.4. Evaluacijske metrike

Za evaluaciju i komparaciju raznih modela za procjenu dubine uobičajeno je korištenje pet evaluacijskih metrika: apsolutna relativna pogreška (Abs Rel), kvadratna relativna pogreška ($\mathbf{Sq\ Rel}$), drugi korijen srednje kvadratne pogreške ($\mathbf{RMSE}$), drugi korijen srednje kvadratne pogreške s obzirom na logaritam dubine ($\mathbf{RMSE\ log}$) te točnost ($\mathbf{Accuracy}$):

$$\mathbf{Abs\ Rel} = \frac{1}{|N|}\sum_{i \in N}\frac{|d_i - d_i^*|}{d_i^*} \tag{4.16}$$

$$\mathbf{Sq\ Rel} = \frac{1}{|N|}\sum_{i \in N}\frac{||d_i - d_i^*||^2}{d_i^*} \tag{4.17}$$

$$\mathbf{RMSE} = \sqrt{\frac{1}{|N|}\sum_{i \in N}||d_i - d_i^*||^2} \tag{4.18}$$

$$\mathbf{RMSE\ log} = \sqrt{\frac{1}{|N|}\sum_{i \in N}||\log(d_i) - \log(d_i^*)||^2} \tag{4.19}$$

$$\mathbf{Accuracy} = \frac{1}{|N|}\sum_{i \in N}[\![\max(\frac{d_i}{d_i^*}, \frac{d_i^*}{d_i}) < \delta]\!] \tag{4.20}$$

U svim gornjim jednadžbama, d_i označava procjenu dubine za i-ti piksel, d_i^* točan iznos dubine za i-ti piksel, N je skup svih piksela dubinske mape, a $|N|$ njihov ukupan broj. Metrika (4.20) računa udio piksela čija je procijenjena vrijednost dubine pogrešna za manje od faktora δ. Ta se metrika može računati za više vrijednosti δ. U sklopu ovog rada, koriste se tri kategorije točnosti $a1$, $a2$ i $a3$ za odgovarajuće δ, δ^2 i δ^3, uz $\delta < 1.25$

5. Eksperimenti

Svi eksperimenti provedeni su s *monodepth2* modelom. Svaka modifikacija modela u odnosu na izvorni, ukoliko postoji, posebno je istaknuta u opisu unutar svakog eksperimenta. Dodatno, objašnjeno je na koji način je model modificiran i što je drugačije u odnosu na izvorni *monodepth2* model. Uporaba hiperparametara koji se razlikuju od standardnih je također istaknuta. Za svaki eksperiment koji se odnosi na učenje ili ugađanje modela te evaluaciju na skupu KITTI, dobiveni rezultati su prikazani tablicom. Eksperimenti su provedeni na NVIDIA GeForce RTX 2080 Ti grafičkoj kartici, s memorijom veličine 11 GB. Također, korišteno je virtualno okruženje *miniconda*, uz biblioteku za duboko učenje PyTorch 1.0.0 te inačicu programskog jezika Python 3.6.6.

5.1. Korišteni skupovi podataka

5.1.1. Skup podataka KITTI

Skup podataka KITTI [2] prikupljen je tijekom šestosatne automobilske vožnje unutar te u okolici grada Karlsruhe u Njemačkoj. Pritom su korišteni razni senzori; PointGray Flea2 kamere (po dvije sive i dvije RGB) za prikupljanje video podataka, Velodyne 3D rotirajući laserski skener za prikupljanje dubine na kojima se algoritmi mogu evaluirati te GPS/IMU inercijski navigacijski sustav visoke preciznosti za pripupljanje lokalnih i globalnih referentnih položaja. Podaci su kalibrirani, sinkronizirani te sadrže vremenske oznake. Nizovi slikovnih okvira dostupni su u rektificiranom te "sirovom" obliku. Skup podataka sastoji se od raznovrsnih scenarija; obuhvaća reprezentativne prometne situacije te okolinu koja varira od autoceste i ruralnih krajeva do unutrašnjosti grada, uz pojavu mnoštva statičnih i dinamičnih objekata. Skup podataka objavljen je u svrhu poticanja razvoja algoritama u području računalnog vida i robotike usmjerenih na autonomnu vožnju. U sklopu ovog rada, za eksperimente učenja i ugađanja koristi se KITTI eigen_zhou podjela, koja se sastoji od 39,810 primjera za učenje te 4,424 primjera za

validaciju. Za konačnu evaluaciju modela koristi se testni skup KITTI Eigen koji se sastoji od 697 primjera. Prilikom evaluacije modela na testnom skupu, procijenjene dubinske mape skaliraju se na rezoluciju 1242×375 na kojoj se računaju evaluacijske metrike. Udaljenost između parova kamera (osnovica) na vozilu iznosi 0.54 m, no ta se vrijednost ne koristi za učenje monokularne procjene dubine - jedino ukoliko se učenje provodi (i) nad stereo podacima.

5.1.2. Skup podataka BIH

Skup podataka BIH je neoznačen, interni skup podataka. Prikupljen je koristeći GoPro Omni opremu, koja se sastoji od 6 HERO4 Black kamera. Podaci su prikupljeni vožnjom po cestama Federacije Bosne i Hercegovine. Slike od kojih se sastoji skup podataka predstavljaju prednji pogled kojeg je generirala jedna od 6 kamera. Video sarži 25 slikovnih okvira po sekundi, na rezoluciji 2704×2028. Predpripremom podataka, slike su skalirane na rezoluciju 384×288. Za navedenu kameru[1], iz podataka proizvođača[2] mogu se očitati parametri horizontalnog i vertikalnog vidnog polja kamere, potrebnih za aproksimaciju intrinsične matrice navedene kamere. BIH skup za učenje sastoji se od 171,477 primjera $\{I_{t-1}, I_t, I_{t+1}\}$. BIH validacijski skup sastoji se od 5,654 primjera. U eksperimentima se za potrebe učenja te ugađanja modela koristi i slučajni podskup BIH_50k koji se sastoji od 50,000 trojki te slučajni validacijski podskup od 700 trojki. S obzirom da je skup podataka BIH neoznačen, učinci učenja i ugađanja modela ispituju se kvalitativno, korištenjem slučajnog podskupa BIH_val_100 koji se sastoji od 100 slika iz BIH validacijskog skupa.

5.2. Korišteni hiperparametri

Za svaki eksperiment, podrazumijeva se da je korišteni model naučen uz hiperparametre dane u ovom odjeljku. Korištenje drugačijih vrijednosti hiperparametara za pojedine eksperimente istakunto je u njihovim pripadajućim odjeljcima.

[1] https://community.gopro.com/t5/en/Available-Video-Resolutions-for-HERO4-Black-and-Silver/ta-p/394196

[2] https://community.gopro.com/t5/en/HERO4-Field-of-View-FOV-Information/ta-p/390285

Tablica 5.1: Hiperparametri modela

Naziv	Opis	Vrijednost
h	Visina slike za učenje/evaluaciju	192
w	Širina slike za učenje/evaluaciju	640
$Optim$	Optimizacijski algoritam	$Adam$
(β_1, β_2)	Koeficijenti gibajućih prosjeka za Adam	$(0.9, 0.999)$
bs	Veličina mini-grupe	12
α	Stopa učenja	10^{-4}
n	Broj epoha	20
γ	Faktor propadanja koraka učenja	0.1
k	Broj epoha za faktor propadanja γ	15
$Arch$	Arhitektura modela	$resnet$
l	Broj slojeva	18
$Skup\,podataka$	Skup podataka za učenje/validaciju	$KITTI$
$KITTI\,podjela$	Podjela korištena za učenje	$eigen\_zhou$
λ	Težina gubitka glatkosti	10^{-3}
μ	Težina gubitka fotometrijske rekonstrukcije	1
s	Broj skala za učenje	4
d_{min}	Najmanja dubina	0.1
d_{max}	Najveća dubina	100
n_{pose}	Broj ulaznih slika za PoseCNN	2
$Stereo$	Korištenje stereo parova tijekom učenja	✗
$ImageNet\,init$	Korištenje ImageNet inicijalizacije	✓

5.3. Reproduciranje modela monodepth2

Model *monodepth2* reproduciran je uz zadane hiperparametre (5.1) i evaluacijske metrike (4.4), na jednoj grafičkoj kartici.

Tablica 5.2: Reproducirani i izvorni monodepth2

Model	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
$monodepth2$	0.115	0.902	4.862	0.193	0.877	0.959	0.981
$monodepth2^*$	0.115	0.896	4.835	0.193	0.877	0.959	0.980

U gornjoj tablici, $monodepth2^*$ označava reproducirani model, a $monodepth2$ je model preuzet iz odgovarajućeg repozitorija na GitHub-u[3]. Model je uspješno reproduciran, a taj rezultat validira ispravnost programskog i sklopovskog okruženja za preostale eksperimente. Reproducirani model je naučen uz hiperparametar $bs = 14$ (5.2), što je mogući uzrok malčice boljih vrijednosti za metrike Sq Rel i RMSE.

5.4. Intrinsična matrica kamere za skup BIH

Intrinsična matrica kamere (2.10) sudjeluje u postupku nenadziranog učenja modela za monokularnu procjenu dubine kroz izraz (4.1). Intrinsična matrica za skup KITTI je unaprijed poznata i dostupna zahvaljujući autorima. Da bi se ispoštovala geometrija, tijekom učenja/ugađanja na skupu KITTI koristi se intrinsična matrica K_{kitti}, a za učenje/ugađanje na skupu BIH koristi se K_{bih}, čije su žarišne duljine aproksimirane formulom:

$$f = \frac{\omega}{2 \tan \frac{\theta}{2}} \tag{5.1}$$

Koristeći izraz (5.1), aproksimiraju se f_x i f_y. Pritom, za f_x, ω je širina slike u pikselima, a θ horizontalno vidno polje u stupnjevima. Analogno, za f_y, ω je visina slike u pikselima, a θ vertikalno vidno polje u stupnjevima. Intrinsična matrica je oblika:

$$K_{bih} = \begin{bmatrix} f_x & & p_x \\ & f_y & p_y \\ & & 1 \end{bmatrix} \tag{5.2}$$

[3]https://github.com/nianticlabs/monodepth2

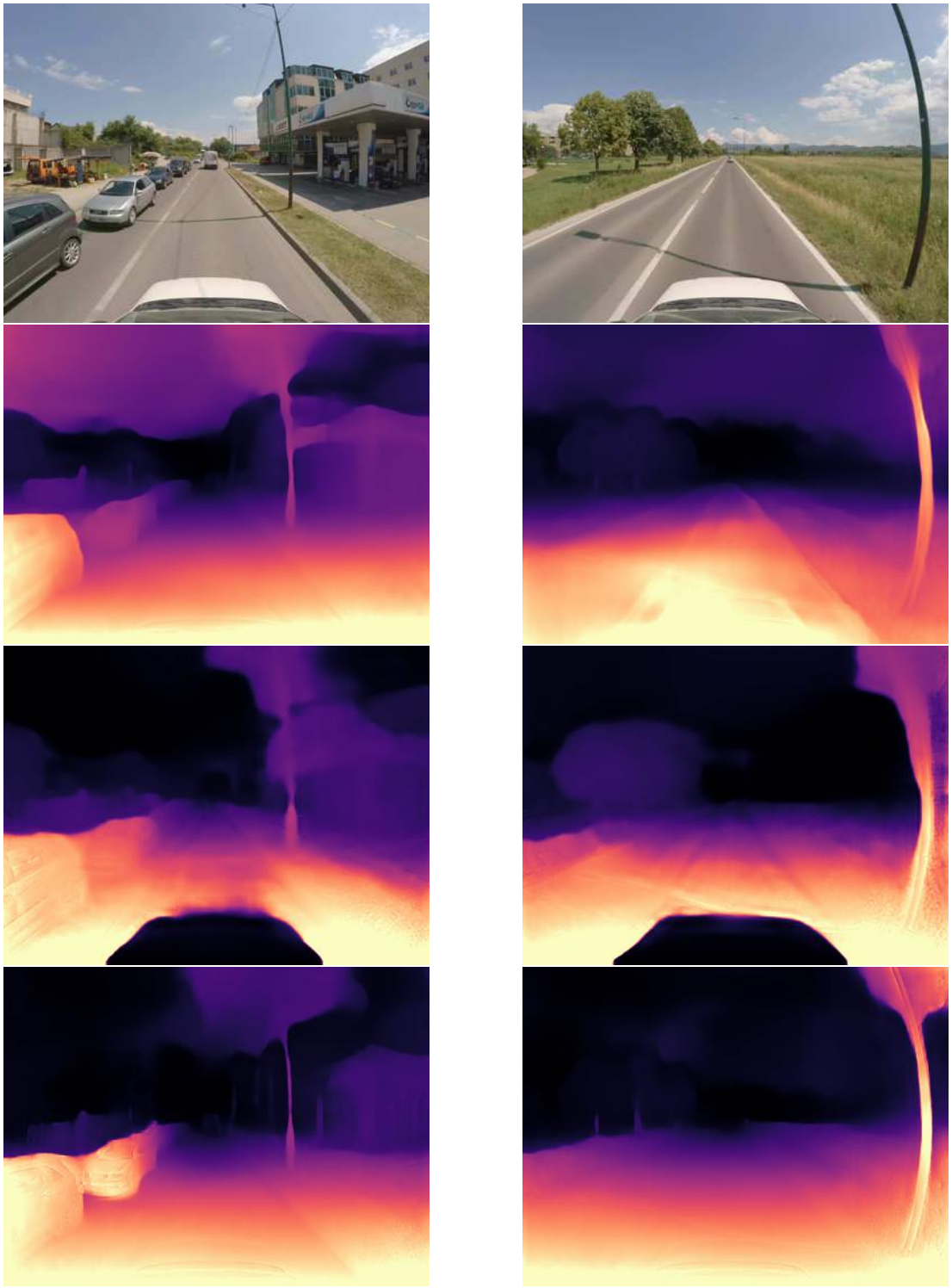
U programskoj implementaciji, K_{bih} se inciijalizira na normaliziranu vrijednost $K_{bih,norm}$ dobivenu upotrebom (5.1) i normalizacijom, a koja glasi:

$$K_{bih,norm} = \begin{bmatrix} 0.27 & & 0.5 \\ & 0.46 & 0.5 \\ & & 1 \end{bmatrix} \tag{5.3}$$

Normalizirana matrica $K_{bih,norm}$ izračunata je pomoću izraza (5.1) za $\omega = 1$. Prije početka prve epohe učenja modela, matrica K_{bih} se inicijalizira na $K_{bih,norm}$, a zatim se njene "prave" vrijednosti dinamički izračunaju, ovisno o rezoluciji koju se koristi za učenje modela. Nadalje, p_x i p_y inicijaliziraju se na $p_x = p_y = 0.5$ jer se pretpostavlja da se glavna točka p nalazi na sredini slikovne ravnine. x-komponente normalizirane intrinsične matrice se pomnože vrijednošću širine ulazne slike, a y-komponente vrijednošću visine. Korištenje normalizirane matrice u programskoj implementaciji je praktično jer f_x i f_y ovise o rezoluciji na kojoj se obavlja učenje.

5.5. Kvalitativno ispitivanje učinaka učenja i ugađanja

U ovom odjeljku, ručno se ispituju predikcije različith inačica modela na nekoliko ulaznih primjera iz validacijskog skupa podataka BIH_val_100 te skupa podataka BIH_50k, s ciljem boljeg razumijevanja učinaka učenja i ugađanja modela. Pritom je od posebne važnosti učinak učenja/ugađanja na različitim skupovima podataka (BIH_50k, KITTI) koji su drugačije distribucije, snimani različitim kamerama te unose specifične pristranosti u model tijekom učenja. Evaluacija performansi modela na validacijskom skupu BIH_val_100 provodi se "od oka", budući da je skup podataka BIH neoznačen.



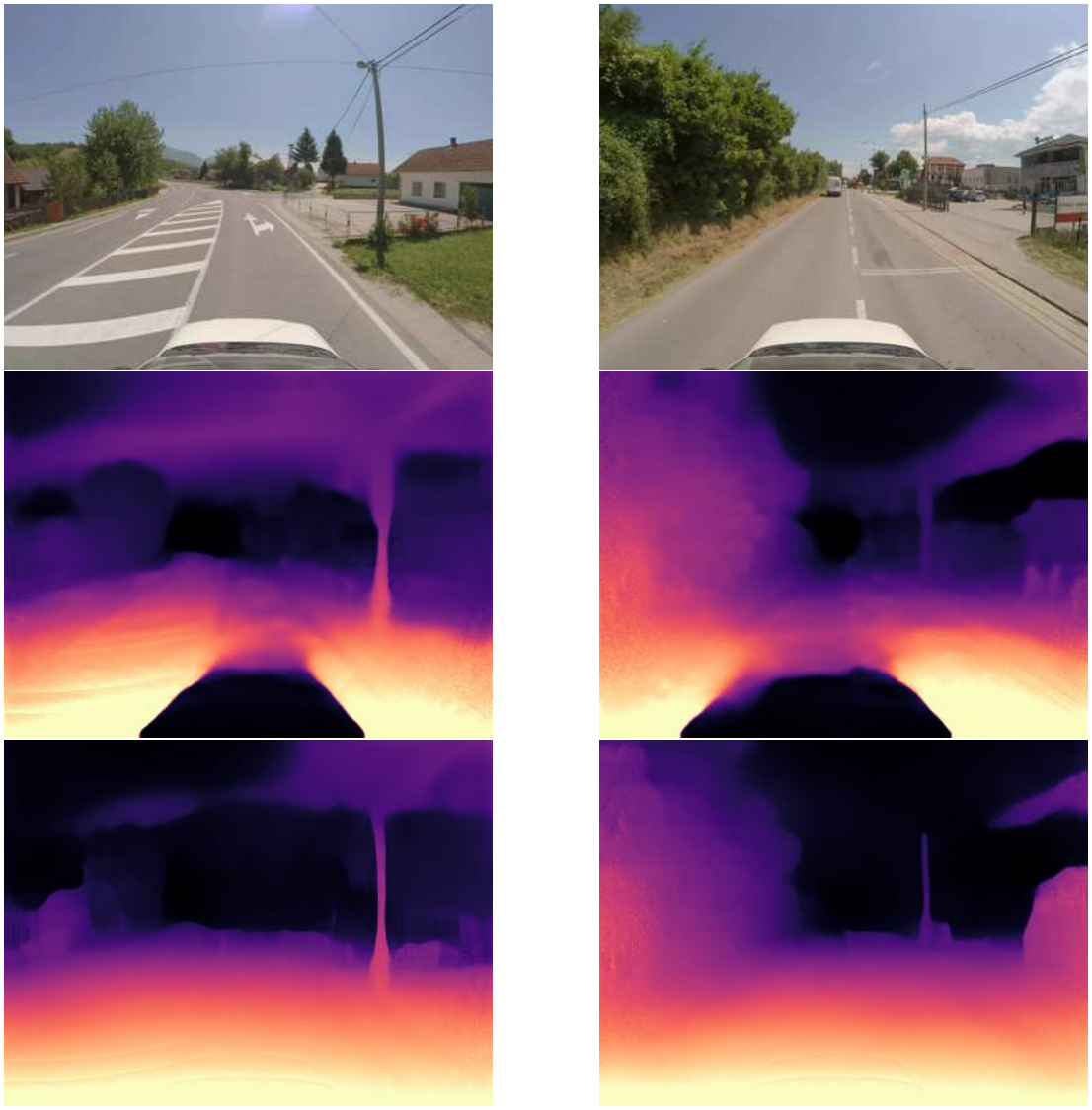
Slika 5.1: Dva primjera iz skupa BIH_val_100 (prvi redak). Predikcije modela monodepth2 učenog i ugađanog na različitim skupovima podataka (ostali retci).

Prvi redak slike 5.1 prikazuje dva primjera iz skupa BIH_val_100. Na desnoj slici vidljiv je stup svjetlosne rasvjete koji je zaobljen jer iz skupa podataka BIH nisu uklonjena radijalna izobličenja.

Drugi redak slike 5.1 prikazuje mape dispariteta koje su izlaz reproduciranog modela monodepth2 (5.3). Neke od pogrešaka koje su lako primjetive odnose se na nebo (posebno na lijevoj mapi) te područje "krošnje" u okolini vrhova stupova ulične rasvjete. Treći redak slike 5.1 prikazuje model monodepth2 naučen na skupu BIH_50k, uz rezoluciju 640×416. Razlika koju se momentalno može primijetiti u odnosu na predikcije u drugom retku slike jest crni auto. Naime, skup podataka BIH prikupljen je na način da je prednji kraj auta vidljiv u svakom slikovnom okviru, na istoj poziciji, kao na slikama u prvom retku. Činjenica da je prednji kraj auta crn u mapama dispariteta, implicira da je model naučio "beskonačnu" udaljenost za taj dio slike - auto je crn koliko i nebo. Iako je to pogreška, budući da se prednji kraj auta nalazi vrlo blizu kamere (za razliku od neba), takav rezultat je smislen s teoretskog gledišta. Naime, za svaki primjer $\{I_{t-1}, I_t, I_{t+1}\}$ tijekom učenja, prednji kraj automobila je objekt zanemarivog dispariteta, a model je upravo to i naučio - objekt je konstantan kroz slikovne okvire - malen disparitet $\rightarrow$ velika dubina. Donošenje teoretski smislenih odluka od strane modela imponira, ali je u ovom slučaju pogrešno, iako model "ima opravdanje". Jedno moguće rješenje je ukloniti takve sveprisutne statične objekte iz skupa za učenje, idealno već u sklopu prikupljanja podataka (npr. pozicionirati kameru na način da se prednji kraj automobila ne vidi). Drugo moguće rješenje je osloniti se u potpunosti na maskiranje (4.3.5). Problem maskiranja je što bez obzira na isključivanje piksela nepovoljnih za procjenu transformacije relativne poze iz funkcije gubitka, model svejedno mora generirati procjene dubine za navedene piksele. Pitanje je što usmjerava postupak učenja modela za procjenu dubine na takvim područjima, ako se maskiranjem takvi pikseli isključuju iz optimizacijskog postupka. S druge strane, procjena dubine u području neba je znatno bolja u odnosu na predikcije u drugom retku slike, vjerojatno zbog toga što je gornji dio KITTI slika (gdje se obično nalazi nebo) podrezan, a na skupu BIH nebo je prisutno u gotovo svim slikama, što je modelu omogućilo prilagodbu. Tendencija "krošnjastih" predikcija na vrhovima tanjih, vertikalno izduženih objekata (ulična rasvjeta) je ponovno pristuna, kao i na predikcijama u drugom retku slike. Četvrti redak slike 5.1 prikazuje mape dispariteta koje je generirao model monodepth2 prednaučen na skupu podataka BIH_50k, a zatim ugađan na KITTI. Model je učen na rezoluciji 640×416. Ta rezolucija donekle odgovara odnosu širine i visine slika iz BIH skupa podataka. Širina slike je jednaka standardnoj za KITTI (5.2), no omjer širine i visine slike od KITTI nije očuvan. Efekt predučenja na BIH_50k, a zatim ugađanja na KITTI je vidljiv; dispariteti u području neba su znatno točniji u odnosu na predikcije u trećem retku slike, a mape dispariteta djeluju finije i oštrije. Međutim, model ponovno griješi na području u okolini vrhova ulične rasvjete na obje slike. Prema

mapama dispariteta, dalo bi se netočno pretpostaviti da su stupovi zapravo drveća s okruglim krošnjama. Teško je sa sigurnošću tvrditi zašto model donosi takve odluke u tim područjima, a iz evaluacijskih metrika takve pogrešne tendencije modela također nisu vidljive, budući da su te metrike makro ocjene. Međutim, može se pretpostaviti da je modelu kroz skupove za učenje, posebice kroz skup BIH, često prikazivano drveće s okruglim krošnjama - predikcije u navedenim područjima su manifestacija (pre)naučenosti modela na specifičnosti jednog (ili oba) od skupova podataka.

5.6. Učinak ugađanja monodepth2_bih modela na skupu KITTI



Slika 5.2: Usporedba predikcija monodepth2_bih i monodepth2_bih+kitti modela

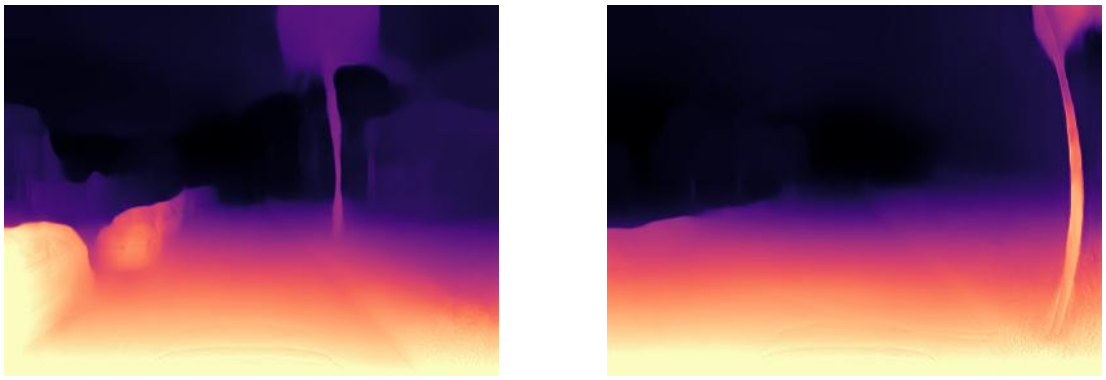
Slika 5.2 prikazana je u tri retka. U prvom se nalaze dvije slike iz skupa BIH. Srednji redak prikazuje predikcije monodepth2_bih modela, naučenog na skupu BIH_50k. Posljednji redak prikazuje predikcije monodepth2_bih+kitti modela, prednaučenog na skupu BIH_50k, a potom ugađanog na KITTI. Zanimljivo je promotriti učinke ugađanja modela na skupu KITTI, za model kojeg se kvalitativno evaluira na skupu BIH. Prema svemu što je vidljivo na slici 5.2 te ranijim eksperimentima iz ovog odjeljka, moguće je izvući nekolicinu objedinjujućih zaključaka po pitanju predučenja na skupu BIH_50k te potom ugađanja na skupu KITTI. Kao i na slici **??**, monodepth2_bih model predviđa "dubinsku rupu" za "statični" prednji kraj automobila. Prikazane dvije slike skupa BIH odabrane su, između ostalog, jer sadrže stupove ulične ravjete i dalekovode, a prema mapama dispariteta očigledno je da model znatno griješi u tim regijama. S obzirom na evaluaciju na skupu BIH_val_100, učinci ugađanja na skupu KITTI jasno su vidljivi. Prije svega, mape dispariteta monodepth2_bih+kitti djeluju finije i oštrije. Drugo, zanimljiva je pojava "invertiranja" predikcije za regiju prednjeg dijela automobila. Na skupu KITTI prednji kraj automobila nije vidljiv, a model je ugađanjem naučio da se prednji kraj automobila nalazi u blizini kamere. Potencijalno objašnjenje invertiranja te smanjenja pogreške dispariteta ceste u okolini vozila je sljedeće; jedna od važnijih vizualnih značajki na koje se modeli za monokularnu procjenu dubine oslanjaju je vertikalna pozicija objekta na slici [18]. Drugim riječima, učenjem, model nauči da svijet ispred kamere "stoji na ravnoj plohi", pod određenim kutem u odnosu na pravac koji izlazi iz kamere, odnosno da postoji visoka korelacija između visine na kojoj se objekt pojavljuje na slici te njegove udaljenosti od kamere. Kod monodepth2_bih modela, maskiranje (4.3.5) koje izvodi monodepth2 isključuje većinu piksela koji pripadaju automobilu i cesti. Dinamika učenja dovodi model do toga da predikcija u tim regijama budu takve kakve jesu - dubinska rupa za automobil, premaleni dispariteti za određene segmente ceste koji su zapravo u blizini kamere. Međutim, kod ugađanja na KITTI koje rezultira monodepth2_bih+kitti modelom - prednjeg kraja automobila na slikama nema, a maska (4.3.5) manje isključuje cestu u odnosu na skup BIH. Drugim riječima, prilikom ugađanja, svi ti pikseli ulaze u optimizacijski postupak koji u konačnici dovodi do toga da je navedenim regijama čija je vertikalna pozicija relativno malena potrebno pridružiti manju dubinu u odnosu na monodepth2_bih inicijalizaciju jer to naprosto smanjuje gubitak. Osim toga, monodepth2_bih+kitti zadržao je poboljšanja za regije na nebu nastale učenjem na BIH, u odnosu na učenje isključivo na KITTI gdje su te pogreške mnogo izraženije. Također, ugađanje drastično smanjuje pogrešku na dalekovodima, a spomenuta tendencija "krošnjaste" predikcije preostaje (lijeva slika).

U tablici 5.3, D_p označava skup za predučenje, a D_t skup za učenje/ugađanje. Model na kojeg se odnosi prvi redak slike je učen uz standardne hiperparametre. Model koji odgovara drugom retku tablice učen je na BIH_50k skupu, uz rezoluciju 640×416 te veličinu mini-grupe 7. Posljednji redak tablice prikazuje model predučen na BIH_50k skupu te ugađan na skupu KITTI, uz rezoluciju 640×416 i veličinu mini-grupe 7. Poboljšanja su vidljiva u tablici. Pogoršanje kvadratne relativne pogreške govori da postoje predikcije koje više odskaču nego ranije, što u kvadratnoj metrici dolazi do izražaja. Svi modeli su evaluirani skupu KITTI, na rezoluciji na kojoj su učeni.

Tablica 5.3: Evaluacija modela monodepth2 s obzirom na učenje/ugađanje na različitim skupovima podataka

D_p	D_t	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
✗	KITTI	0.115	0.902	4.862	0.193	0.877	0.959	0.981
✗	BIH_50k	0.215	1.930	7.379	0.313	0.669	0.877	0.942
BIH_50k	KITTI	0.108	0.933	4.768	0.188	0.890	0.962	0.981

5.7. Povećanje težine gubitka glatkosti



Slika 5.3: Predikcije monodepth2_bih+kitti modela s većom težinom gubitka glatkosti

Slika 5.3 prikazuje mape dispariteta monodepth2_bih+kitti modela, ugađanog uz veću težinu gubitka glatkosti $\lambda = 2 \cdot 10^{-2}$, koja je za faktor 20 veća u odnosu na standardnu vrijednost tog hiperparametra (5.2). Ako se gornje slike usporede s 5.1, vidi se da je ugađanje uz veći λ utjecalo na tendenciju "krošnjastih" predikcija, tj. pogreška je manja u tim regijama. Međutim, pojavljuju se veće i teže uočljive pogreške u drugim regijama. Analizom "od oka" može se na nekim mjestima uočiti povećanje dubine na

5.3 u odnosu na 5.1 (npr. drugi auto s lijeva, nebo, itd.). Brojke donekle idu tome u prilog; medijan omjera medijana svih "istinitih" i procijenjenih dubinskih mapa se povećao. Evaluacijske metrike se nisu znatno promijenile - što također ide u prilog tome da je model negdje dobio, a negdje izgubio na kvaliteti procjene.

Tablica 5.4: Evaluacija modela s obzirom na λ

λ	Med	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
10^{-3}	30.696	0.108	0.933	4.768	0.188	0.890	0.962	0.981
$2 \cdot 10^{-2}$	31.343	0.108	0.931	4.782	0.188	0.889	0.961	0.981

Tablica 5.4 pokazuje evaluaciju monodepth2_bih+kitti modela, ugađanog uz dvije različite vrijednosti hiperparametra λ. Evaluacija je provedena na skupu KITTI. Postoji vrlo malena razlika u performansama modela s obzirom na 7 standardnih evaluacijskih metrika. Med označava medijan omjera medijana svake "istinite" i procijenjene dubinske mape. Povećanje navedenog medijana s povećanjem težine glatkosti za faktor 20 moglo bi značiti da s porastom težine glatkosti i srednja vrijednost dubine raste - ukoliko dubine promatramo kao slučajne varijable normalne distribucije. Ipak, za potvrdu te hipoteze potrebna je evaluacija većeg broja modela učenih s različitim vrijednostima hiperparametra λ.

5.8. Interpolacija transponiranom konvolucijom

Da bi model na izlazu generirao dubinske mape koje su iste rezolucije kao i ulazne slike (3.4), dekoder provodi naduzorkovanje međureprezentacija. U tu svrhu, model monodepth2 koristi interpolaciju najbližim susjedom. S druge strane, jedna od prirodnih metoda naduzorkovanja u konvolucijskim arhitekturama je *transponirana konvolucija*, često zvana i *dekonvolucija*. Međutim, ne radi se o operaciji koja je stvarni matematički inverz konvolucije i koja uklanja "efekt konvolucije". Transponirana konvolucija se tako naziva jer se zaista radi o transponiranoj matrici konvolucije. Ako se ulaz u konvoluciju, npr. 4×4, organizira (s lijeva na desno, odozgo prema dolje) u vektor, konvoluciju se može svesti na množenje ulaza rijetkom matricom $\boldsymbol{C}$ (gdje npr. nastaje izlaz 2×2) čiji su ne-nul elementi težine konvolucijskog filtera $\omega_{i,j}$ (gdje je i redak, a j stupac konvolucijskog filtera). Koristeći takvu reprezentaciju, kod unazadnog prolaska, odnosno unazadne propagacije pogreške, gubitak se množi s $\boldsymbol{C}^T$. Ta operacija prima izlaz konvolucije kao vektor 4×1 te na svom izlazu daje vektor 16×1 kojega se na isti način može preurediti u 4×4. Dakle, neka konvolucijski filter $\boldsymbol{\omega}$ definira

matricu konvolucije $\boldsymbol{C}$ za koju se unaprijedni i unazadni prolazak računaju množenjem s $\boldsymbol{C}$, odnosno $\boldsymbol{C}^T$. Taj filter ujedno definira i transponiranu konvoluciju čiji se unaprijedni i unazadni prolazak računaju množenjem s $\boldsymbol{C}^T$, odnosno $(\boldsymbol{C}^T)^T = \boldsymbol{C}$. Dakle, transponiranu konvoluciju zgodno je promatrati kao operaciju koja nastaje zamjenom unaprijednog i unazadnog prolaska konvolucije. U implementaciji modela monodepth2, u slojevima dekodera reprezentacije se naduzorkuju za faktor 2 (jer se u koderu konvolucijom 3×3 s pomakom 2 poduzorkouju za isti faktor). Pritom se kao metoda naduzorkovanja koristi interpolacija najbližim susjedom. Modelu se može omogućiti da nauči interpolaciju koja mu je (lokalno) optimalna, uvođenjem transponirane konvolucije. S obzirom da je interpolacija najbližim susjedom utjecala samo na prostorne dimenzije reprezentacija, zamijenjena je transponiranom konvolucijom koja ne mijenja broj kanala C, tj. broj ulaznih i izlaznih kanala je jednak. Nadalje, koriste se konvolucijski filter dimenzija 3×3 (jer se takav koristi kroz cijeli model), uz pomak 2 koji za transponiranu konvoluciju konceptualno odgovara umetanju nula između elemenata ("usporava" filter, obrnuto od konvolucije) te nadopunjavanje. S navedenim parametrima postiže se naduzorkovanje reprezentacije za faktor 2, uz nepromijenjen broj kanala. Takvom konvolucijom nadomješta se interpolacija najbližim susjedom.

Tablica 5.5: Usporedba modela s obzirom na metodu naduzorkovanja u dekoderu

Naduzorkovanje	Rezolucija	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
interpolacija naj. sus.	640×192	0.115	0.902	4.862	0.193	0.877	0.959	0.981
interpolacija naj. sus.	640×416	0.107	0.844	4.659	0.185	0.889	0.963	0.982
transp. konv.	640×416	0.106	0.819	4.585	0.183	0.893	0.963	0.982

Standardni model monodepth2 izvodi interpolaciju najbližim susjedom u dekoderu. Prema rezultatima u tablici 5.5 vidljivo je da zamjena interpolacije odgovarajućim transponiranim konvolucijama daje bolje rezultate, što pokazuje da je model naučio naduzorkovanje korisnije od interpolacije najbližim susjedom. Model s transponiranom konvolucijom učen je na rezoluciji 640×416, uz veličinu mini-grupe 7. Učenje nastavlja zauzimati nešto manje od 11GB memorije na grafičkoj kartici. Model prikazan u srednjem retku tablice je standardni monodepth2, učen uz iste hiperparametre kao model s transponiranom konvolucijom. Zanimljivo, glavnina pomaka u rezultatima zapravo dolazi od promjene hiperparametra rezolucije, no vidljivo je da transponirana konvolucija ipak daje bolje rezultate u odnosu na interpolaciju najbližim susjedom.

5.9. Aleatorna nesigurnost

Teoretsko i praktično znanje primijenjeno u ovom eksperimentu proizlazi iz radova [11], [19] i [12]. Nesigurnost se u dubokim modelima, bilo za regresiju ili klasifikaciju, može izraziti kroz *Bayesovsko duboko učenje*. Dvije su osnovne vrste nesigurnosti - *aleatorna* i *epistemička*. Epistemička nesigurnost je "nesigurnost modela", tj. odnosi se na nesigurnost parametara modela - koji je model generirao priukupljene podatke? Epistemička nesigurnost "nestaje" uz dovoljno velik skup za učenje. Epistemička nesigurnost nije predmet ovog eksperimenta. S druge strane, aleatorna nesigurnost odnosi se na podatke - obuhvaća šum koji je inherentan podacima. Primjerice, šum senzora (kamera, laser, IMU...), šum gibanja, itd. Prema tome, aleatorna nesigurnost ne smanjuje se s porastom skupa podataka. Nadalje, aleatorna nesigurnost dijeli se na *homoskedastičku* i *heteroskedastičku* nesigurnost. Homoskedastička nesigurnost je konstantna s obzirom na ulazne podatke, ne mijenja se. Heteroskedastička nesigurnost ovisi o ulaznim podacima, može se mijenjati za različite podatke na ulazu. Primjerice, kod monokularne procjene dubine, očekivano je da predikcije na regijama bogate teksture budu male nesigurnosti. Analogno, za predikcije na regijama siromašne teksture, objekte koji se gibaju, visoko reflektirajuće površine te površine koje ne zadovoljavaju Lambertov zakon - očekuje se visoka nesigurnost procjene. U sklopu ovog eksperimenta implementirana je *heteroskedastička aleatorna nesigurnost*.

Ideja je sljedeća; ukoliko se u model $\hat{y} = f(X;\theta)$ ugradi procjena nesigurnosti, on postaje $(\hat{y}, \sigma) = f(X;\theta)$, gdje je $\hat{y}$ procijenjena dubinska mapa, σ je procijenjena nesigurnost, $f(\cdot)$ funkcija modela, θ parametri modela, a X ulazni podaci. Veličinu σ procjenjuje model, tj. neuronska mreža koja procjenjuje i mape dispariteta. Budući da je neuronska mreža funkcija ulaznih podataka i vlastitih parametara, posjeduje slobodu prilagođavanja procjene nesigurnosti različitim ulaznim podacima. Prema probabilističkoj formulaciji, neka su izlazi neuronske mreže $(\hat{y}, \sigma)$ parametri posteriorne distrubucije $p(y|\hat{y}, \sigma)$, gdje y predstavlja oznake. Korištenjem Laplaceove distribucije:

$$p(y|\hat{y}, \sigma) = \frac{1}{2\sigma} \exp \frac{-|y - \hat{y}|}{\sigma} \tag{5.4}$$

Tijekom učenja, cilj je minimizacija negativne log izglednosti koja proizlazi iz distribucije (5.4):

$$-\log p(y|\hat{y}, \sigma) = \frac{|y - \hat{y}|}{\sigma} + \log \sigma + konst. \tag{5.5}$$

U kontekstu modela za monokularnu procjenu dubine, L1 norma u (5.5) zapravo je jedna od komponenata gubitka fotometrijske rekonstrukcije (4.11). To je ujedno i razlog zašto se u ovom eksperimentu koristi Laplaceova distribucija, umjesto, primjerice,

Gaussove (koja je kompatibilna s L2 normom). Dakle, L1 norma između ciljne slike I_t i njene rekonstrukcije, kao komponenta funkcije gubitka, pretvara se u izraz (5.5). Procjena nesigurnosti svodi se na procjenu mape nesigurnosti, na način da je u sve konvolucijske slojeve dekodera koji generiraju mape dispariteta dodan još jedan konvolucijski filter za generiranje mape nesigurnosti. Prema tome, mape nesigurnosti generiraju se na svim skalama kao i mape dispariteta te se jednako tako naduzorkuju na izvornu rezoluciju prije izračuna funkcije gubitka. Također, u sklopu eksperimenta oprobane su dvije varijante; u jednoj se koristi sigmoida kao aktivacijska funkcija i nesigurnosti su tada $\sigma \in [0, 1]$, a u drugoj se izlaz navedenih konvolucijskih slojeva (nema aktvacijske funkcije) tretira kao $s_i = \log \sigma$ te se tada izraz (5.5) mijenja u:

$$-\log p(y|\hat{y}, \sigma) = \exp(-s_i)|y - \hat{y}| + s_i \tag{5.6}$$

Izraz (5.6) jednak je izrazu (5.5), uvrsti li se s_i. Međutim, dinamika učenja je drugačija (vrijednosti funkcije gubitka), a rezultati se također razlikuju. Dodatno, u izrazu (5.5), zbog numeričke stabilnosti koristi se $\delta = 10^{-6}$ pri dijeljenju i izračunu prirodnog logaritma.

Tablica 5.6: Usporedba modela monodepth2 učenih s aleatornom nesigurnosti i standardnog modela bez maskiranja

Akt. f-ja	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
✗	0.120	1.097	5.074	0.197	0.872	0.956	0.979
✗	0.135	1.591	5.254	0.209	0.856	0.953	0.980
sigmoida	0.132	1.534	5.271	0.205	0.860	0.954	0.980

Prvi redak tablice 5.6 prikazuje evaluaciju standardnog modela monodepth2, učenog bez maskiranja (4.3.5). Drugi i treći redak prikazuju modele učene s aleatornom nesigurnosti te identičnim hiperparametrima, uz spomenute razlike u implementaciji koje se odnose na interpretaciju izlaza konvolucijskih slojeva dekodera te (ne)korištenje aktivacijske funkcije. Razlika u hiperparametrima između standardnog modela i modela učenih s aleatornom nesigurnosti je broj epoha - potonji su učeni 30 epoha, profitirali su od duljeg učenja uz standardne hiperparametre. Nadalje, za učenje modela s aleatornom nesigurnosti nije korišteno monodepth2 maskiranje (4.3.5) - ideja je bila nadomjestiti maskiranje modeliranjem aleatorne nesigurnosti te vidjeti što je model naučio apostrofirati u svojim mapama nesigurnosti. Prema rezultatima evaluacije na skupu KITTI prikazanim u tablici, učenje s aleatornom nesigurnosti nije unaprijedilo

predikcije dubine. Međutim, mape nesigurnosti donose novu, korisnu informaciju, koju model prethodno nije imao.

Slika 5.4: Primjeri iz skupa KITTI

Gornje slike pripadaju KITTI validacijskom skupu, model ih nije vidio tijekom učenja.

Slika 5.5: Mape nesigurnosti za primjere 5.4

Slika 5.5 prikazuje mape nesigurnosti koje je generirao naučeni model (treći redak) iz tablice 5.6, s prethodno opisanim postavkama i hiperparametrima. Model procjenjuje visoku nesigurnost na rubovima objekata gdje se javlja diskontinuitet dubine. Također, na obje slike je visokom nesigurnošću označen objekt koji se giba (biciklist). Na desnoj slici, također se ističe visoka nesigurnost za pojedine visoko reflektirajuće površine (dijelovi automobila, prozori). Aleatorna nesigurnost je vrlo korisna informacija - može se iskoristiti prilikom dodatne optimizacije, kao što je to učinjeno u [19].

5.10. Rekonsturkcija značajki

U prijašnjim eksperimentima, model je učen rekonstrukcijom ciljne slike iz okolnih (4.13). Pritom su za rekonstrukciju korištene dubinske mape iz više slojeva dekodera, naduzorkovane bilinearnom interpolacijom na rezoluciju ulaza, uz minimum (4.12) na svakoj rezoluciji te u konačnici, prosjek po korištenim skalama. Uz uzastopne slike, imamo pristup i njihovim značajkama, tj. reprezentacijama koje nastaju u skrivenim slojevima kodera i dekodera arhitekture U-Net za procjenu dubine. Prema tome, u postupak učenja možemo uvesti dodatno ograničenje - rekonstrukciju značajki. Moguće je implementirati rekonstrukciju značajki bez modifikacija u postojećoj arhitekturi modela monodepth2. Imajući u vidu geometriju koja omogućuje nenadzirano

učenje (4.1), to je moguće iz dva razloga; procjena relativne poze kamera, tj. transformacija $\hat{T}_{t\to s}$ (2.1.6), procjenjuje se i djeluje na razini koordinatnog sustava, a značajke su dimenzija (N, C, H, W) kao i ulazne slike, uz veći broj kanala i manje prostorne dimenzije. Nadalje, uz izlazni sloj, dubinsku mapu procjenjuju i 3 skrivena sloja dekodera iz modula za procjenu dubine. Ako se rekonstruiraju značajke iz ta 3 skrivena sloja, kao dubinu tih značajki možemo iskoristiti dubinske mape koje ti slojevi procjenjuju. Dimenzije značajki skrivenih slojeva i dimenzije dubinskih mapa koje ti slojevi procjenjuju se poklapaju, budući da se dubinska mapa dobiva primjenom konvolucije 1×1 nad navedenim značajkama.

Dakle, za rekonstrukciju značajki, izraz (4.1) poprima oblik:

$$\Phi(p_s) \sim K\hat{T}_{t\to s}\hat{D}_t(\Phi(p_t))K^{-1}\Phi(p_t) \tag{5.7}$$

U gornjem izrazu funkcija $\Phi(\cdot)$ označava djelovanje (de)kodera na ulaznu sliku, tj. reprezentaciju koja se rekonstruira. Gubitak rekonstrukcije značajki računamo isto kao fotometrijski gubitak (4.5):

$$L_{fr} = \sum_s \sum_p |\Phi(p_t) - \Phi(p_s)| \tag{5.8}$$

Gornji izraz, kojim se dobiva gubitak L_{fr}, superponira se ukupnom gubitku fotometrijske rekonstrukcije (4.13), uz težinu λ_{fr}. Važno je istaknuti nekoliko detalja u vezi logike samog eksperimenta. U izrazu (5.7), korištena je transformacija $\hat{T}_{t\to s}$ koja se računa samo jednom za svaki par $\{I_t, I_s\}$, odnosno ne računa se posebna transformacija za značajke. U dekoderu modula za procjenu dubine, za svaki skriveni sloj (rezoluciju) čije se značajke koriste za rekonstrukciju, već postoje odgovarajuće dubinske mape $\hat{D}_t$. Konačan gubitak rekonstrukcije značajki (5.8) je minimum po vremenskoj dimenziji (4.12), kao kod rekonstrukcije ciljnog pogleda.

Tablica 5.7 prikazuje naučene 3 inačice modela monodepth2, uz gubitak (5.8), fiksan parametar λ_{fr} te različit broj skrivenih slojeva s dekodera čije se značajke koriste za rekonstrukciju. Zbog memorijskih ograničenja, korištena je veličina mini-grupe 10. Rezultati u tablici pokazuju da je najbolji model učen uz rekonstrukciju značajki iz 3 skrivena sloja dekodera. Taj model je ujedno i nešto bolji od standardnog modela monodepth2 (odjeljak 5.3), ali nedovoljno da bismo to smatrali unaprijeđenjem modela.

Tablica 5.7: Evaluacija modela monodepth2 učenog uz rekonstrukciju značajki dekodera i korištenje različitog broj slojeva za rekonstrukciju

λ_{fr}	s	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
0.01	3	0.115	0.904	4.836	0.192	0.878	0.959	0.981
0.01	2	0.116	0.898	4.840	0.195	0.875	0.958	0.980
0.01	1	0.116	0.895	4.848	0.195	0.874	0.958	0.980

5.11. Miješanje uz pomoć sloja pažnje

Preskočne veze između odgovarajućih slojeva kodera i dekodera su arhitekturalne značajke koje pospješuju performanse modela kod zadataka guste predikcije. Iz tog razloga se koriste i u konvolucijskoj arhiktekturi korištenoj za monokularnu procjenu dubine. Značajke iz slojeva kodera dovode se izravno do odgovarajućih slojeva dekodera preskočim vezama i konkateniraju se s njima po dimenziji kanala, dok prostorne dimenzije moraju biti jednake. Značajke kodera omogućuju slojevima dekodera da uzmu u obzir lokalni kontekst prilikom izgradnje globalne strukture (3.4). U implementaciji, radi se o konkatenaciji uz jediničnо preslikavanje značajki kodera u dekoder:

$$H = concat(H_{dec}, IH_{enc}) \tag{5.9}$$

Umjesto jediničnog preslikavanja značajki kodera IH_{enc} u dekoder, modelu možemo alocirati kapacitet za učenje proizvoljne funkcije preslikavanja koja optimizacijom možda može postati korisnija modelu. Uporabom sloja pažnje za učenje proizvoljne funkcije, u model unosimo novu induktivnu pristranost - tranformacija značajki kodera ovisi o značajkama dekodera. Drugim riječima, modelu omogućujemo da s obzirom na značajke dekodera, nauči pogodnu transformaciju značajki kodera - po mogućnosti bolju od jediničnog preslikavanja, inače alocirani kapacitet nije pretjerano djelotvoran. Izraz (5.9) postaje:

$$H = concat(H_{dec}, attention(H_{dec}, H_{enc})) \tag{5.10}$$

Model prikazan u drugom retku tablice 5.8 učen je s miješanjem uz pomoć sloja pažnje. Prvi redak prikazuje standardni model monodepth2. Modeli su učeni s istim hiperparametrima, osim rezolucije 448×288 te veličine mini-grupe 6 uz koje je učen model s miješanjem uz pomoć pažnje. Rezultati su osjetno slabiji od standardnog modela, ali postoji mogućnost da se bolji rezultati mogu dobiti odabirom prikladnije

Tablica 5.8: Evaluacija modela monodepth2 s miješanjem uz pomoć sloja pažnje

Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
0.115	0.902	4.862	0.193	0.877	0.959	0.981
0.140	1.251	5.640	0.218	0.829	0.942	0.976

stope učenja, strategije ažuriranja stope učenja i korištenjem "zagrijavanja" u optimizacijskom postupku za parametre koji pripadaju sloju pažnje.

6. Zaključak

Iako se radi o loše postavljenom problemu, model za monokularnu procjenu dubine moguće je izvesti te nenadziranim učenjem naučiti na neoznačenim snimkama. Nenadzirano učenje modela za monokularnu procjenu dubine temeljeno na rekonstrukciji pogleda pokazuje se kao obećavajući pristup koji, za razliku od nadziranog učenja, nije ograničen oznakama. Nije isključeno da takav sustav u budućnosti nadjača i stereo sustave te da i drugi zadaci u području računalnog vida profitiraju od znanja koje takav sustav posjeduje. Iako se kroz postupak rekonstrukcije pogleda dubina uči neizravno te iako postoje dvije duboke neuronske mreže u sustavu tijekom učenja – jedna za procjenu relativne poze kamera te jedna za procjenu dubine, rezultat učenja je model za monokularnu procjenu dubine koji funkcionira samostalno. Informacija o relativnoj pozi između kamera postaje nepotrebna jer se na ulaz modela dovodi jedan pogled, a na izlazu model generira odgovarajuću dubinsku mapu. Prostor za napredak postoji jer je sustav visokog stupnja kompleksnosti i sastoji se od nekoliko komponenata, a zadatak je sam po sebi vrlo zahtjevan. Postavlja se pitanje kako adresirati činjenicu da model pretpostavlja statičnost svijeta, odnosno da primjeri iz kojih model uči o svijetu proizlaze isključivo iz gibanja kamere. Činjenica je da se svijet u lokalnom okruženju kamere može gibati te da pretpostavka o njegovoj statičnosti nije uvijek točna. Navedena pretpostavka ima znatan utjecaj na sposobnost generalizacije modela te uzrokuje pojave poput, primjerice, „crnih rupa u dubinskoj mapi". Ako u model nije ugrađen mehanizam za detekciju nezavisnog gibanja u sceni, taj se problem ne može kvalitetno adresirati. Srodni problemi, poput kamere koja miruje ili objekata koji se gibaju poput kamere, uglavnom se rješavaju maskiranjem odgovarajućih piksela. Međutim, maskiranje piksela nije idealno rješenje. Primjer je eksperiment učenja modela monodepth2 na skupu BIH_50k, a potom učinak ugađanja na skupu KITTI. Nadalje, model za procjenu relativne poze kamera u modelu monodepth2 gotovo se u potpunosti oslanja na učenje s kraja na kraj i velik kapacitet dubokog modela. Drugim riječima, nema ugrađene induktivne pristranosti koje bi mu dodatno pomogle da kapitalizira nad činjenicom da su dvije ulazne slike uzastopne, tj. da se vjerojatno velikim dijelom

preklapaju, a da se pritom najviše informacija o pomaku kamere nalazi u dijelovima koji se ne preklapaju ili su pomaknuti. Moguće je da bi ugradnja te pretpostavke kroz odgovarajuće mehanizme ili integracija s modelom za, primjerice, optički tok, pomogla u rješavanju tih prepreka. Nije jasno koji je smjer najbolji. Također, tipična funkcija gubitka korištena za nenadzirano učenje monokularne procjene dubine vrlo je složena. Određena proširenja, poput indeksa strukturalne sličnosti i regularizacijskog gubitka pospješuju generalizaciju modela. Moguće je da postoji bolja opcija od L1 norme na razini piksela koja je osjetljiva na korespondentne piksele čija se vrijednost mijenja u uzastopnim slikama. Nadalje, podaci su izazovni sami po sebi. Primjerice, površine koje ne zadovoljavaju svojstvo Lambertove refleksije stvaraju probleme jer se njihova prividna svjetlina mijenja za različite kuteve gledanja. Također, kroz vremenske intervale, svjetlina se mijenja - modelu je potrebna robusnost na fluktuacije svjetline u sceni. Nadalje, pitanje je postoji li univerzalno i elegantno rješenje za probleme poput zaklanjanja i artefakata nastalih kopiranjem teksture. Izvor tih problema je također u podacima, ali moguće je da unaprijeđenje korištenih tehnika i arhitektura, bilo na razini dubokog učenja ili specifičnih za monokularnu procjenu dubine, doprinese ili čak eliminira utjecaj takvih pojava na generalizaciju modela. Moguće je da postoje bolje, (ne)otkrivene neuralne arhitekture koje su prikladnije za monokularnu procjenu dubine te procjenu relativne poze kamera u odnosu na trenutno korištene i poznate.

LITERATURA

[1] Pinhole camera. URL https://en.wikipedia.org/wiki/Pinhole_camera.

[2] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[3] C. Godard, O. M. Aodha, and G. J. Brostow. Digging into self-supervised monocular depth estimation. *CoRR*, abs/1806.01260, 2018. URL http://arxiv.org/abs/1806.01260.

[4] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003. ISBN 0521540518.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. 2016.

[8] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015.

[9] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. 2016.

[10] R. Jain, R. Kasturi, and B. Schunck. *Machine Vision*. 01 1995. ISBN 978-0-07-032018-5.

[11] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? 2017.

[12] M. Klodt and A. Vedaldi. Supervising the new with the old: learning sfm from sfm. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[13] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. 2015.

[14] O. Ozyesil, V. Voroninski, R. Basri, and A. Singer. A survey of structure from motion. 2017.

[15] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. 2015.

[16] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 2010. ISBN 1848829345.

[17] K. Tateno, F. Tombari, I. Laina, and N. Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. 2017.

[18] T. van Dijk and G. C. H. E. de Croon. How do neural networks see depth in single images? 2019.

[19] N. Yang, L. von Stumberg, R. Wang, and D. Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. 2020.

[20] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. 2017.

Sažetak

Monokularna procjena dubine loše je postavljen problem kojeg karakterizira nedostatak geometrijskih ograničenja, a primjena dubokog učenja pokazuje se plodonosnom u računalnom vidu. Pristup zadatku podrazumijeva nenadzirano učenje dubokih konvolucijskih neuronskih mreža za procjenu dubine i relativne poze kamere, temeljenih na široko rasprostranjenoj arhitekturi U-Net. Pojedine induktivne pristranosti te korisna geometrijska svojstva - poput rijetke povezanosti i ekvivarijantnosti s obzirom na pomak, razlog su zbog kojeg spomenute arhitekture predstavljaju prikladan alat za rješavanje navedenog zadatka. Matematički okvir za nenadzirano učenje monokularne procjene dubine proizlazi iz modela kamere i geometrije nastanka slike, a svodi se na zadatak rekonstrukcije. Kod monokularnog pristupa, posebno se ističe problem skale te potreba za modelom za procjenu relativne poze kamera - dva povezana problema kojih nema kod stereo pristupa. Kvalitativni rezultati pokazuju pojedine učinke učenja i ugađanja modela na skupu KITTI te na internom skupu BIH. Arhitekturalne značajke poput procjene aleatorne nesigurnosti, rekonstrukcije značajki i mehanizma pozornosti, pokazuju se kao kompetitivne. Eksperimenti također pokazuju da se performanse modela monodepth2 na KITTI Eigen testnom skupu mogu unaprijediti pažljivijim odabirom vrijednosti pojedinih hiperparametara te upotrebom transponirane konvolucije za naduzorkovanje. Svi eksperimenti su temeljeni na modelu monodepth2.

Ključne riječi: monokularna procjena dubine,nenadzirano učenje,konvolucijska neuronska mreža,duboko učenje,model rupičaste kamere

Title

Abstract

Monocular depth estimation is an ill-posed problem that suffers from lack of geometric constraints, whilst deep learning has already proven itself for a variety of computer vision tasks. The approach consists of training deep convolutional neural networks deployed as popular and widely used U-Net architecture. These tools are known for possessing desirable inductive biases (e.g. sparse connectivity) and geometric properties (e.g. translational equivariance). Unsupervised learning for monocular depth estimation is enabled by a mathematical framework that arises from single view geometry and the pinhole camera model; the neural networks are trained to perform warping-based view synthesis. The "scale problem" and the need for relative camera pose estimation model are the key differences between monocular and stereo approach. Qualitative results show the effects of training and fine-tuning on KITTI and internal BIH datasets. Architectural features such as aleatoric uncertainty estimation, multi-scale feature reconstruction and attention mechanism are also competitive. Experiments also show that monodepth2 performance on KITTI Eigen split can be improved by choosing different hyperparameter values and using transposed convolution for upsampling. All experiments are based on monodepth2 model.

Keywords: monocular depth estimation,unsupervised learning,convolutional neural networks,deep learning,pinhole camera model