

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 482

Korespondencijska ugrađivanja za stereoskopsku rekonstrukciju

David Kerman

Zagreb, svibanj 2022.

ZAVRŠNI ZADATAK br. 482

Pristupnik: **David Kerman (0036522014)**

Studij: Elektrotehnika i informacijska tehnologija i Računarstvo

Modul: Računarstvo

Mentor: prof. dr. sc. Siniša Šegvić

Zadatak: **Korespondencijska ugrađivanja za stereoskopsku rekonstrukciju**

Opis zadatka:

Stereoskopska rekonstrukcija važan je zadatak računalnog vida s mnogim zanimljivom primjenama. Tom problemu možemo pristupiti ugrađivanjem piksela slike u visokodimenzionalni metrički prostor koji je pogodan za ostvarivanje korespondencije. Ugrađivanje možemo naučiti na stereoskopskim slikama s poznatom dubinskom informacijom. U okviru rada, potrebno je odabrati okvir za automatsku diferencijaciju te upoznati biblioteke za rukovanje tenzorima i slikama. Proučiti i ukratko opisati postojeće pristupe za stereoskopsku rekonstrukciju koji se temelje na dubokom učenju. Odabrati slobodno dostupni skup slika te oblikovati podskupove za učenje, validaciju i testiranje. Predložiti prikladnu arhitekturu dubokog modela za korespondencijsko ugrađivanje. Uhodati postupke učenja modela i validiranja hiperparametara. Primijeniti naučene modele te prikazati i ocijeniti postignutu točnost. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 10. lipnja 2022.

Zahvaljujem mentoru prof. dr. sc. Siniši Šegviću, na savjetima i ukazanoj pomoći. Zahvaljujem kolegi Antoniu Ilinoviću na diskusijama i pomoći tijekom izrade rada. Također, zahvaljujem svojoj obitelji, prijateljima i kolegama na potpori tijekom mog dosadašnjeg obrazovanja.

SADRŽAJ

1. Uvod	1
2. Stereoskopska rekonstrukcija	2
2.1. Epipolarna geometrija	2
2.2. Pretprocesiranje slika	3
2.2.1. Rektifikacija	4
2.3. Disparitet	4
2.4. Algoritam stereoskopske rekonstrukcije	6
3. Duboko učenje	8
3.1. Osnovni pojmovi u dubokom učenju	8
3.1.1. Perceptron	8
3.1.2. Prijenosne funkcije	9
3.1.3. Konvolucijski slojevi	12
3.2. Učenje dubokih modela	13
3.2.1. Funkcija gubitka	14
3.2.2. Gradijentni spust	14
3.2.3. Optimizacija	15
3.2.4. Regularizacija	15
4. Podatkovni skup za učenje	17
4.1. Podatkovni skup KITTI 2015	17
5. Duboko učenje u kontekstu stereoskopske rekonstrukcije	19
5.1. Ugrađivanje okana u metrički prostor	19
5.2. Učenje modela za ostvarivanje korespondencije	21
6. Eksperimentalni rezultati	23

7. Programska izvedba i vanjske biblioteke	26
8. Zaključak	27
Literatura	28

1. Uvod

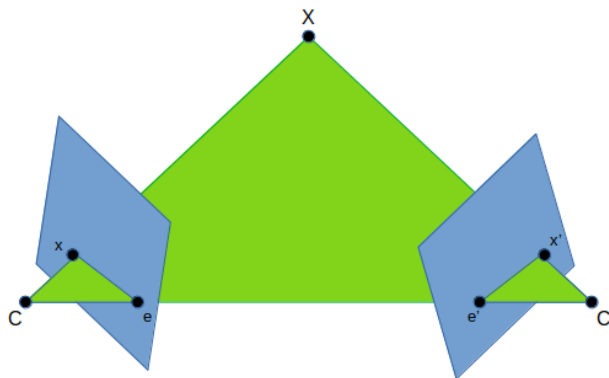
Računalni vid kao jedna od temeljnih grana umjetne inteligencije omogućava računalima da izluče značajnu informaciju iz slika, videa ili sličnih vizualnih ulaza. Područje stereoskopske rekonstrukcije ima za cilj ostvariti stvaran trodimenzionalan položaj točaka koje su promatrane u dvije ili više slika, te kao takvo ima vrlo široke primjene uključujući one za određivanje sličnosti slikovnih okana. Kroz povijest postojale su klasične metode za određivanje korespondencijskih metrika, no pojavom snažnijih računala iskorišteno je duboko učenje kao alat kojim je moguće korespondencijske metrike naučiti na stvarnim podacima. Za potrebe metoda temeljenih na ugrađivanju u visokodimenzionalni prostor potrebne su rektificirane stereoskopske slike s poznatom dubinskom informacijom.

U okviru ovog radu opisana je geometrija stereoskopskog para, kalibracija, rektifikacija i algoritmi korišteni za ostvarivanje korespondencije. Zatim definirani su osnovni pojmovi u dubokom učenju, te postupci korišteni u dubokom učenju uz memorijske zahtjeve. Predstavljen je podatkovni skup na kojem su izvršeni postupci treniranja i validacije. Opisan je model kojim je ostvareno korespondencijsko ugrađivanje slikovnih okana, koji su korišteni u stereoskopskoj rekonstrukciji. Nastavno na to, opisan je postupak učenja takvog modela, te eksperimentalno vrednovanje rekonstrukcijske točnosti.

2. Stereoskopska rekonstrukcija

2.1. Epipolarna geometrija

Epipolarna geometrija je geometrija koja ima primjenu u stereoskopskoj rekonstrukciji, a po definiciji opisuje odnos točaka u slikama dobivenih iz dviju kamera. Također naziva se i epipolarno ograničenje jer uvelike smanjuje broj točaka koje je potrebno provjeriti pri traženju korespodentnih točaka. Točka X koja je prikazana na slici 2.1 prikazuje točku u 3D prostoru koja se snima iz dviju kamera. Točke C i C' prikazuju centre lijeve, odnosno desne kamere. Projekcija točke X na ravninu lijeve kamere je x , a na ravninu desne kamere x' . Važno je napomenuti da su točke X , x i x' u istoj ravnini, te zajedno sa centrima kamera C i C' čine tzv. *epipolarnu ravninu* π .



Slika 2.1: Epipolarna ravnina u kojoj se nalaze točka X , te točke nastale projekcijom x i x'

Nadalje, na slici su označeni *epipolovi* oznakama e i e' te oni prikazuju točke u kojima pravac koji povezuje centre kamera C i C' siječe slikovne ravnine. Najvažniji element su *epipolarne linije*, označene dužinama koje spajaju x i e , te x' i e' , ali su zapravo pravci koji su nastali presjekom epipolarne ravnine sa slikovnom ravninom. Epipol je točka u kojoj se sijeku svi epipolarni pravci.

Poznavajući činjenicu da se točke nalaze u istoj ravnini, te ako znamo položaj točke x možemo odrediti položaj točke x' . Rješenja koja se moraju ispitati da bi saznali položaj točke x' nalaze se na epipolarnoj liniji, što uvelike olakšava pronalaženje korespondentnih točaka.

2.2. Pretprocesiranje slika

Parametre geometrije sustava kamera moguće je podijeliti na dvije vrste: intrinzični i ekstrinzični. Ekstrinzični parametri su oni koji opisuju odnos svojstva koja su zajednička kamerama. Intrinzični pak opisuju svojstva kamera koje se odnose svaku kameru pojedinačno. Kao primjere za intrinzične faktore navodimo svojstva koja utječu na nji, a to su nesavršenost leća (na slici 2.2 prikazano je radijalno izobličenje), pomak senzora od centra leća te slične fizičke karakteristike nesavršenosti kamera.



Slika 2.2: Primjeri radijalnog izobličenja. Izvor:[2]

S druge strane ekstrinzični parametri nas dovode do nužnih transformacija kojima se slike dovode u istu ravninu projekcije te se time postiže da su pikseli duž horizontalnog pravca jedne slike na istoj visini na drugoj slici. Taj pravac je već spomenut, riječ je o epipolarnoj liniji. Ovakvim transformacijama uvelike se olakšava pronalaženje korespondentnih točaka koje u ovom slučaju treba tražiti samo duž jedne linije, epipolarne linije. Problem je sveden na jednu dimenziju. Kalibracijom se rješava problem nesavršenosti leća, odnosno problem izobličenja slika.

2.2.1. Rektifikacija

Za postupak rektifikacije važni su ekstrinzični faktori kamera, dok na njihov izračun utječu intrinzični faktori. Transformacijom slika rektifikacijom dobivamo slike čiji pikseli koji u prostoru odgovaraju točkama na istoj visini, a nalaze se duž iste epipolarne linije. Također, nakon provedenih transformacija intrinzični parametri kamera postaju jednaki.



Slika 2.3: Prikaz epipolarnih linija u rektificiranim slikama skupa KITTI 2015

Postupak se provodi kroz nekoliko koraka, a uključuje šahovsku ploču koja je pogodna zbog svojih svojstava. Prvi korak je traženje šahovskih polja. Postupak se nastavlja određivanjem izobličenja i ostalih parametara uz pomoć šahovskih polja. Na slici 2.2 prikazan je par slika iz stereo kamere koji je rektificiran. Slike su iz skupa KITTI 2015, te se na njima vide epipolarne linije. Lako je primjetiti da iste točke u prostoru nalaze na istoj visini, odnosno na istoj epipolarnoj liniji. Ovakvom transformacijom, korespondentne piksele potrebno je tražiti samo na istoj y -koordinati.

2.3. Disparitet

Prijašnjim postupcima kalibracije i rektifikacije omogućen je postupak računanja dispariteta, odnosno postupak računanja koji za svaki piksel koji je nastao lijevom kamerom određuje piksel slike koja je nastala desnom kamerom, ali pomaknutog za udaljenost d piksela po horizontalnoj osi. *Disparitet* je definiran kao horizontalni pomak d između dvaju korespondentnih piksela u slikama nastalim iz dviju konkurentnih stereo kamera. Kao rezultat računanja dispariteta nastaje mapa dispariteta koja sadrži disparitet za svaki piksel referentne kamere. Odnos piksela slike iz lijeve kamere, koja je u ovom slučaju referentna, i slike iz desne kamere definira se kao:

$$I_L(x, y) = I_D(x - d, y) \quad (2.1)$$

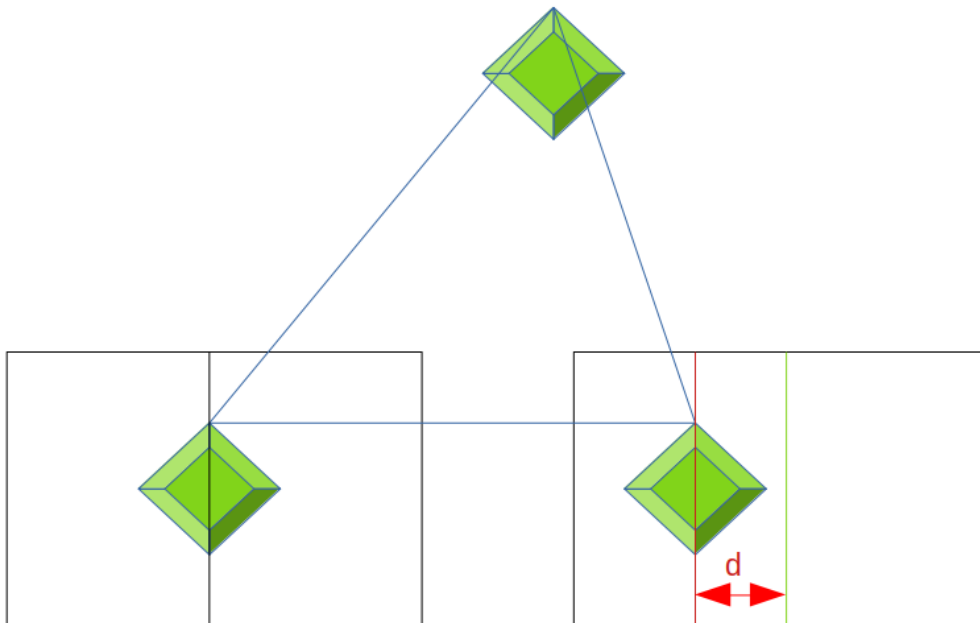
gdje su I_L i I_D slike nastale lijevom, odnosno desnom kamerom, respektivno. Uređenim parom (x, y) opisani su pikseli slike iz lijeve kamere, a njemu korespondentni pikseli

u slikama iz desne kamere označeni su pomakom za disparitet d po horizontalnoj osi $(x - d, y)$. Važno je napomenuti da su slike u skupu KITTI 2015 rektificirane i kalibrirane, što znači da korespondentne piksele tražimo samo po horizontalnoj osi. Pomoću izračunatog dispariteta moguće je izraziti odnos između piksela kamera i dubine scene kao:

$$Z = f \frac{B}{d} \quad (2.2)$$

gdje je dubina scene, odnosno udaljenost objekta od kamere označena sa Z , d je disparitet, B je označena udaljenost između središta kamera, a f je označena fokalna udaljenost kamere. Iz jednadžbe 2.2 je vidljiva obrnuta proporcionalnost između udaljenosti objekta od kamera, te dispariteta s druge strane. Kada je položaj objekta blizu kamere, disparitet će biti velik. A kako se udaljavamo od kamere, disparitet se smanjuje i u beskonačnosti postiže vrijednost 0.

Na slici 2.4. je vidljivo da su epipolarne linije paralelne, te im je y -koordinata za korespondentne piksele jednaka. Disparitet je označen slovom d i prikazuje horizontalni pomak.



Slika 2.4: Prikaz objekta u slikama iz lijeve i desne kamere. Plavom bojom označena je epipolarna ravnina. Okomitim osima označeni su projicirani pikseli u konkurentnim slikama. Disparitet je označen slovom d .

2.4. Algoritam stereoskopske rekonstrukcije

Postupak stereoskopske korespodencije je zapravo postupak pronalaženja istih piksela u međusobno konkurentnim kamerama koji odgovaraju istoj točki u trodimenzionalnom prostoru. Postoje dvije vrste korespodencija, *rijetke* i *guste*. Razlika je u tome što rijetke ne izračunavaju disparitete za sve piksele, dok guste to rade. Guste korespodencije omogućene su današnjom snagom računala, te se rijetke više toliko ne koriste. Postupak guste korespodencije je tim teži što se prilikom izračuna susreće s nekim anomalijama u slikama, primjerice područjima bez tekture, reflesivnim podlogama na kojima se svjetlost drugačije lomi i odbija, te točkama koje se iz jedne kamere vide, a iz druge ne (*stereoskopska sjena*).

Dolazimo do stereo algoritama, svaki od njih moguće je opisati u četiri koraka ([3], [9]):

1. izračunavanje razlike između dvaju slikovnih okana, odnosno izračun podatkovnih cijena po prije definiranoj mjeri razlike
2. prikupljanje podatkovnih cijena za disparitete koji se razmatraju
3. izračun mape dispariteta, odnosno odabir najmanje prikupljene cijene za svaki piksel, uz optimizaciju
4. zaglađivanje mape dispariteta

Prvi korak ovog generičkog algoritma za stereoskopsku rekonstrukciju je izračun podatkovnih cijena. Izračun podatkovnih cijena može se računati prema pojedinačnim pikselima ili skupini piksela. Izračun nam govori kolika je razlika između dvaju slikovnim okana, a razlika koja se koristi je definirana prije. Mjere razlike mogu biti sljedeće:

- srednja kvadratna razlika
- srednja apsolutna razlika
- kvadratna razlika između intenziteta piksela
- apsolutna razlika između intenziteta piksela

Drugi od koraka generičkog stereo algoritma je prikupljanje podatkovni cijena, tj. agregacija. Na početku je postavljena granica za maksimalan disparitet, pa se tijekom ovog koraka izračunava podatkovna cijena za sve disparitete i skupa od 0 do D , te ćemo tako za svaki piksel imati $D + 1$ podatkovnih cijena.

Treći korak algoritma je izračun mape dispariteta. Nakon što su izračunate podatkovne cijene za razmatran skup dispariteta, treba odabrati metodu za izračun mape dispariteta. U praksi postoje *lokalne* i *globalne*. Lokalne metode prikupljaju cijene nad ograničenim područjem unutar same mape dispariteta, odnosno traže na ograničenom području oko referentne točke. Područje može biti 2D ili 3D, te se prikupljanje može primjerice izvesti konvolucijom. S druge strane, globalne metode preskaču korak prikupljanja podatkovnih cijena, te odmah nakon izračuna podatkovnih cijena vrše optimizaciju. Globalne metode minimiziraju kriterij nad cijelom slikom. Odnos između ovi metoda može se svesti na to da su lokalne brže, no globalne daju puno glađe mape dispariteta, što ih čini kvalitetnijima.

Zadnji korak koji je naveden je zaglađivanje mapa dispariteta. Ovaj korak je opcionalan i neke metode ga koriste. Glavni cilj ovog koraka je uklanjanje šuma koji je nastao generiranjem mape.

3. Duboko učenje

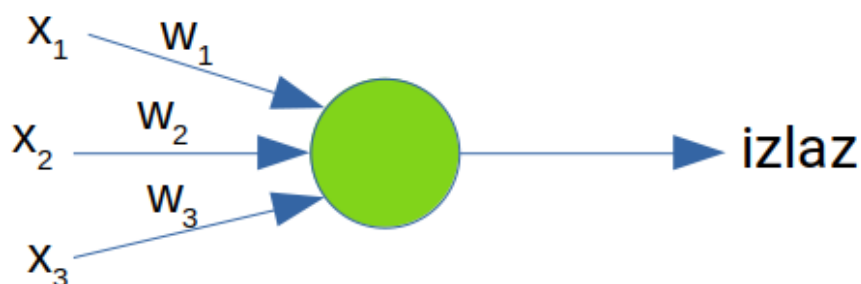
Duboko učenje grana strojnog učenja koje se ubrzano razvija. U teoriji se dubokim učenjem smatra proces učenja neuronske mreže s 3 ili više slojeva, sve što je manje od toga naziva se *plitim* modelima. Cilj takvih neuronskih mreža je simulirati ponašanje ljudskog mozga omogućujući učenje na velikim skupovima podataka. Zahvaljujući razvoju grafičkih kartica, omogućen je ubrzan razvoj dubokog učenja.

U sljedećim odjeljcima bit će predstavljeni osnovni pojmovi i koncepti korišteni prilikom oblikovanja modela i procesa učenja.

3.1. Osnovni pojmovi u dubokom učenju

3.1.1. Perceptron

U ranim počecima duboko učenja došlo se do ideje potpuno povezanih modela. Jedan od takvih modela je višeslojni perceptron. Iako je danas napuštena ideja potpuno povezanih modela, zanimljivo je pogledati koje mane novi modeli imaju za ispraviti.



Slika 3.1: Jednostavni TLU perceptron

TLU perceptron su prvi definirali McCulloch i Pitts. Na slici 3.1. prikazan je jednostavni TLU perceptron. TLU perceptron ima više binarnih ulaza koji su označeni s x_1, x_2 i x_3 , te proizvodi binarni izlaz. Rosenblatt je, kako bi odredio izlaz, definirao težine (u ovom slučaju w_1, w_2 i w_3). Težinama se mogu opisati doprinosi jednog neurona na drugi, odnosno koliko jedan utječe na drugog. Ovime je izlaz neurona definiran sumom umnoška težina i binarnih ulaza te praga s kojim se uspoređuje:

$$\begin{cases} 0, \sum_{k=1}^n w_k x_k \leq \text{prag} & (3.1a) \\ 1, \sum_{k=1}^n w_k x_k > \text{prag} & (3.1b) \end{cases}$$

Jedan sloj neuronske mreže čine perceptroni koji se nalaze u istom stupcu. Izlaz svakog perceptrona izračunava se tako koristeći izlaze prethodnog sloja i definirane težine za svaki od ulaznih perceptrona. Gornju formulu moguće je transformirati tako što se prag uvede nova varijabla $b = -\text{prag}$, koja označava pristranost (engl. *bias*). Pristranost u kontekstu duboko učenja je moguće definirati koliko je lako natjerati neuron da kao izlaz izbaci 1. Ukoliko je pristranost vrlo velik broj, izlaz će u većini slučajeva biti 1, dok je za vrlo negativan broj suprotna situacija. Izmijenjena formula glasi:

$$\begin{cases} 0, \sum_{k=1}^n w_k x_k + b \leq 0 & (3.2a) \\ 1, \sum_{k=1}^n w_k x_k + b > 0 & (3.2b) \end{cases}$$

Linearnu funkciju koju smo izveli možemo predstaviti i u matričnom obliku:

$$f = w^T x + b \quad (3.3)$$

3.1.2. Prijenosne funkcije

Ukoliko bi koristili linearnu funkciju izlaza iz prijašnjeg odjeljka, mala promjena u ulazu bi na izlazu generirala veliku promjenu. Stoga se uvode prijenosne, odnosno aktivacijske funkcije jer odlučuju hoće li neki neuron biti aktivan ili ne. Primarni cilj prijenosnih funkcija je omogućiti da se primijete sitne promjene na ulazu. Aktivaciju neurona možemo predstaviti sljedećom formulom:

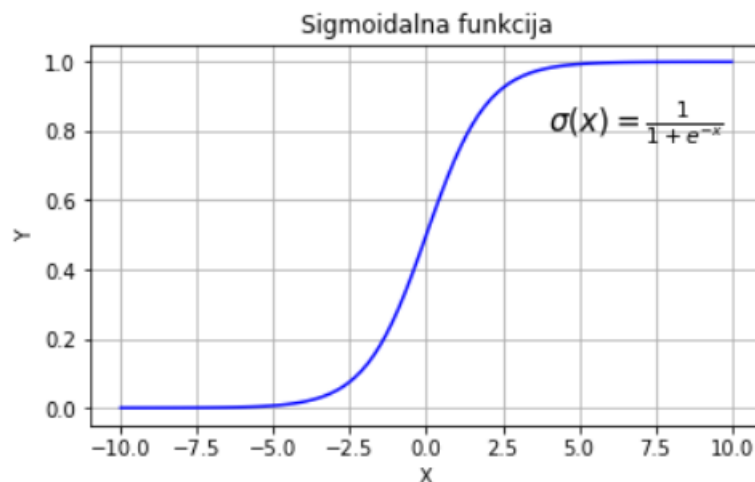
$$a^k = \sigma_k(W_k^T a^{k-1} + b) \quad (3.4)$$

gdje a^k označava aktivaciju neurona trenutnog sloja, a^{k-1} aktivaciju neurona prijašnjeg sloja, dok σ označava aktivacijsku funkciju.

Prva funkcija koja će biti objašnjena je sigmoidalna funkcija čiji izraz je dan u nastavku:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.5)$$

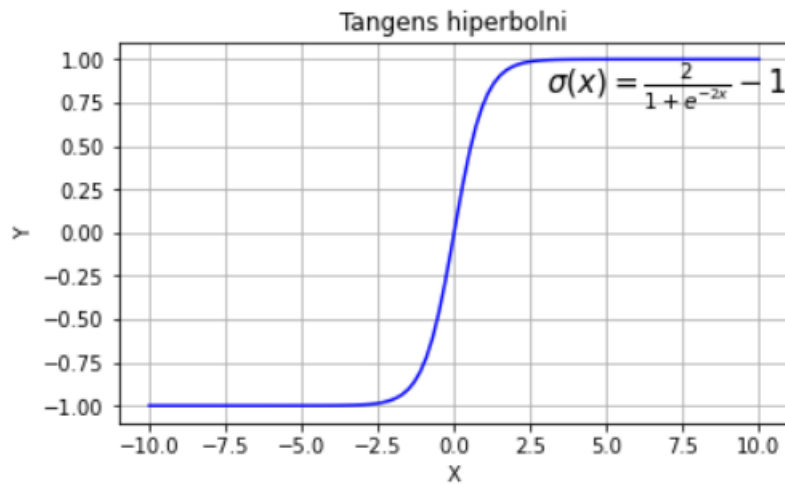
Glavno obilježje sigmoidalne funkcije je to što je njezina kodomena na skupu od 0 do 1, te ju to čini primjenjivom u regresijskim i klasifikacijskim problemima. Veliki pozitivni brojevi poprimat će vrijednost 1, dok će jako negativni brojevi poprimati vrijednost 0. Također, prednost je što je derivabilna, pa omogućuje učenje u kojem se koristi gradijentni spust. Glavni nedostatak zbog kojeg se sve manje koristi je problem nestanja gradijenata (engl. *vanishing gradient problem*) koji govori da gradijenti postaju toliko mali da ne nose nikakvu informaciju koja bi se mogla iskoristiti za učenje. Prikaz grafa funkcije je na slici 3.2.



Slika 3.2: Graf sigmoidalne funkcije

Funkcija koja je usko vezana uz sigmoidalnu je *tangens hiperbolni*. Kodomena ove funkcije je na intervalu od -1 do 1. Problem nestajanja gradijenata je isto tako prisutan. Kako je bliska sigmoidalnoj, možemo uspostaviti odnos između njih koji je oblika:

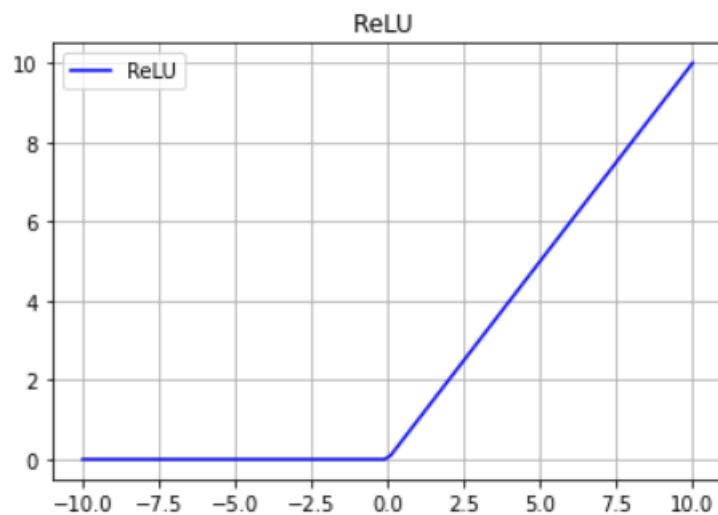
$$\tanh(x) = 2\sigma(2x) - 1 \quad (3.6)$$



Slika 3.3: Graf funkcije tangensa hiperbolnog

Sljedeća prijenosna funkcija koja se koristi je zglobnica (engl. *ReLU* - *rectified linear unit*). Ova prijenosna funkcija propušta pozitivne vrijednosti, dok negativne vrijednosti ne propušta. Glavna prednost ove funkcije što je računski izuzetno jeftina, dok joj je nedostatak što za negativne vrijednosti uvijek daje 0. Izraz koji ju opisuje je oblika:

$$f(x) = \max(0, x) \quad (3.7)$$



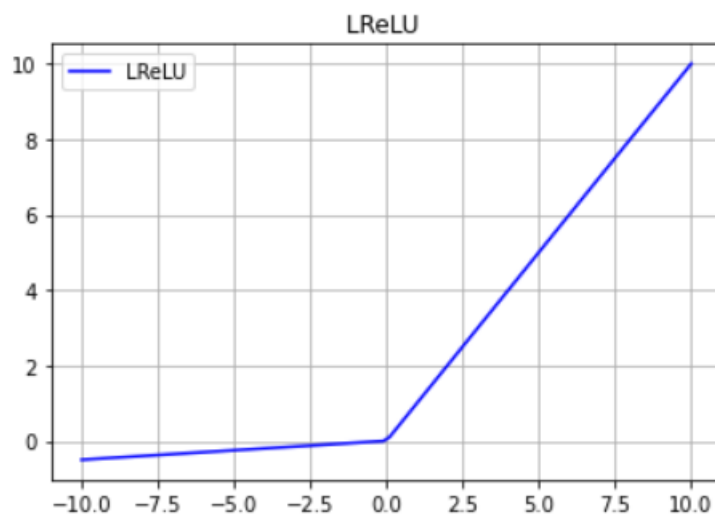
Slika 3.4: Graf funkcije ReLU

Moguće je napraviti modifikaciju zglobnice tako da za negativne vrijednosti daje vrlo male vrijednosti, te se takva prijenosna funkcija naziva propusnom zglobnicom (engl. *LReLU* - *leaky rectified linear unit*). U nastavku slijedi izraz koji ju opisuje:

$$f(x) = \begin{cases} x, & x > 0 \\ \alpha x, & x \leq 0 \end{cases} \quad (3.8a)$$

$$(3.8b)$$

gdje je α parametar, odnosno u praksi vrlo mali pozitivan broj. Na slici 3.5. je vidljivo da negativne vrijednosti propušta uz prigušenje, dok na slici 3.4. obična zglobnica ne propušta.



Slika 3.5: Graf funkcije LReLU

3.1.3. Konvolucijski slojevi

Konvolucijski slojevi osnovni su gradivni element konvolucijskih neuronskih mreža. Kako je u radu riječ o slikama, bit će predstavljena 2D konvolucija, operacija filtriranja. Uloga filtriranja je uočavanje i učenje uzoraka koji se nalaze na slikama. Konvolucijski slojevi čine konvolucije ulaza s različitim filtrima, drugi naziv je jezgra (engl. *kernel*). Izlaz iz 2D konvolucije je aktivacijska mapa, odnosno mapa značajki, dok je izlaz cijelog konvolucijskog sloja više mapa značajki. Kako dubina tijekom prolaska kroz mrežu raste, identificiraju se sve specifičniji uzorci. Tijekom unaprijednog prolaza, jezgra se provlači po matričnoj reprezentaciji slike i računa se skalarni umnožak između jezgre i ulaza na svakom položaju.

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} * \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} = \begin{bmatrix} x_{11}w_{11} + x_{12}w_{12} & x_{12}w_{11} + x_{13}w_{12} \\ +x_{21}w_{21} + x_{22}w_{22} & +x_{22}w_{21} + x_{23}w_{22} \\ x_{21}w_{11} + x_{22}w_{12} & x_{22}w_{11} + x_{23}w_{12} \\ +x_{31}w_{21} + x_{32}w_{22} & +x_{32}w_{21} + x_{33}w_{22} \end{bmatrix} \quad (3.9)$$

Jednadžbom 3.9 prikazana je matricna reprezentacija dimenzija 3×3 , dok je jezgra dimenzije 2×2 . Jezgra se provlači po ulaznoj matrici (računa se skalarni umnožak) i nastaje izlazna mapa značajki dimenzija 2×2 . Iz ovog je vidljivo da se konvolucijom smanjuje dimenzija ulaza.

Ukoliko želimo zadržati rezoluciju slike nakon konvolucijskog sloja, potrebno je uvesti nadopunjavanje (engl. *padding*). Najčešće je korištena metoda popunjavanja nulama. Još jedan od hiperparametara je korak (engl. *stride*). Korak govori o tome koliko se jezgra pomiče u vertikalnom i horizontalnom smjeru. U jednadžbi 3.9 je korišten korak 1.

3.2. Učenje dubokih modela

Učenje dubokih modela, u generalnom slučaju neuronske mreže, je postupak tijekom koje se modelu na ulaz daju podaci iz skupa za učenje. Tijekom procesa učenja model ažurira težine u slojevima s ciljem kako bi što bolje aproksimirao izlaznu funkciju. Za početak kako bi mogli odrediti koliko odstupamo od pravih vrijednosti, uvodimo pojam funkcije gubitka. U procesu učenja provodi se minimizacija funkcije gubitka. Ukoliko na skupu podataka gubitak ispada malen, a na skupu podataka koje još nije vidio ispada izražen, govorimo o tzv. prenaučivosti modela, što znači da model nema sposobnost generalizacije.

Postoji nekoliko pristupa učenju, nadzirano učenje u kojem se modelu predaju parovi oblika (ulaz, željeni izlaz), nenadzirano u kojem se modelu samo predaju ulazni podaci, te podržano učenje u kojem se maksimizira kumulativna nagrada. U ovom radu bit će razmatrano nadzirano učenje.

3.2.1. Funkcija gubitka

Kako bismo procijenili koliko model odstupa od željenih vrijednosti izlaza koristimo funkciju gubitka. Ako model loše procjenjuje, gubitak će biti velik. Kod problema klasifikacije najčešće je korištena unakrsna entropija (engl. *cross-entropy loss*):

$$L_{CE} = - \sum_{i=1}^n t_i \cdot \log(p_i) \quad (3.10)$$

Popularan klasifikator koji koristi unakrsnu entropiju, a daje vjerojatnosnu reprezentaciju izlaza modela je funkcija *softmax*:

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \quad (3.11)$$

3.2.2. Gradijentni spust

U prijašnjem pododjeljku definirana je funkcija gubitka, cilj učenja modela je minimizirati gubitak. Kako bi se parametri pomicali u smjeru minimuma funkcije gubitka, a da istovremeno pomak bude što veći izračunava se gradijent funkcije gubitka.

Neka je zadana funkcija gubitka koja ovisi o težinama \mathbf{W} i pomacima \mathbf{b} . Gradijenti funkcije gubitka mogu se zapisati kao:

$$\nabla L_w = \left(\frac{\partial L}{\partial w_1}, \dots, \frac{\partial L}{\partial w_n} \right) \quad (3.12)$$

$$\nabla L_b = \left(\frac{\partial L}{\partial b_1}, \dots, \frac{\partial L}{\partial b_n} \right) \quad (3.13)$$

gdje n predstavlja broj parametara.

U iteracijama gradijentnog spusta parametri se ažuriraju prema sljedećim izrazima:

$$W_{i+1} = W_i - \epsilon \cdot \nabla L_w(\mathbf{W}, \mathbf{b}) \quad (3.14)$$

$$b_{i+1} = b_i - \epsilon \cdot \nabla L_b(\mathbf{W}, \mathbf{b}) \quad (3.15)$$

gdje je ϵ tzv. stopa učenja (engl. *learning rate*), a predstavljena je malim pozitivnim realnim brojem.[4]

3.2.3. Optimizacija

Kad govorimo o optimizaciji, najkorištenija metoda je svakako gradijentni spust, no korištenjem njega se mogu manifestirati neki problemi koji kao posljedicu imaju nepronalazak optimalnih težina. Postojanje više lokalnih minimuma i spora kovergencija jedni su od problema. U ovom radu je korišten ADAM optimizator, a svakako potrebno je spomenuti i SGD optimizator.

Glavna ideja optimizatora SGD je procijeniti gradijent cijelog skupa uzoraka sa malim slučajno odabranim podskupom uzoraka (mini-grupa, engl. *minibatch*). Već i veličine od 128 daju vrlo dobru aproksimaciju gradijenta. U usporedbi s "običnim" gradijentnim spustom (GD) to znatno ubrzava proces učenja, jer se kod GD izračun treba provesti za cijeli skup. Stohastički gradijentni spust koristi se uz zalet koji ubrzava učenje. Umjesto ažuriranja težina koristeći samo trenutno izračunat gradijent, zalet također uzima u obzir i prethodno ažuriranje, pomnoženo sa faktorom umanjivanja α . [6]

Modifikacija u odnosu na GD je vidljiva sljedećom jednačinom:

$$v = \mu v + \nabla L_w \quad (3.16)$$

,gdje je v trenutno akumuliran zalet, a μ je zalet.

Adaptive Moment Estimation (ADAM) je optimizator koji kombinacijom dvije metodologije gradijentnog spusta zaleta i RMSprop algoritma ubrzava pronalazak globalnog minimuma. Glavna značajka ADAM-a je što koristi prosječno eksponencijalno kretanje gradijenta i kvadratnog gradijenta, te ga to čini izrazito efikasnim na velikim skupovima podataka i u radu s velikim brojem parametara [1].

3.2.4. Regularizacija

Kako bi se izbjegla prenaučenos modela (engl. *overfitting*), odnosno loša generalizacija modela potrebno je uvesti regularizatore koji imaju za cilj spriječiti prenaučenos.

Jedna od metoda koje se koriste je regularizacija dodavanjem regularizacijskog faktora funkciji gubitka. Pri tome treba spomenuti $L1$ i $L2$ regularizaciju, kojima se kažnjava norma vektora težina.

$$L_R = L + \lambda \cdot \sum_i |w_i| \quad (3.17)$$

$$L_R = L + \lambda \cdot \sum_i w_i^2 \quad (3.18)$$

Izrazom 3.17 opisana je $L1$ regularizacija, a 3.18 $L2$ regularizacija.

Regularizacijska metoda koja istovremeno ima i ulogu optimizacije je normalizacija grupe (engl. *batch norm*). Normalizacija grupe primjenjuje se prije nelinearnosti, a temelji se na normalizaciji izlaza na normalnu jediničnu razdiobu. Posljedica takvog pristupa je što ulazi u slojeve imaju sličnu razdiobu, čime se postiže ubrzanje učenja bržom propagacijom gradijenata. U postupku normalizacije u mini grupi za svaku izlaznu mapu značajki računaju se srednja vrijednost i varijanca po sljedećim izrazima:

$$\mu_B = \frac{1}{m} \sum_{i=1}^m \mathbf{x}_i \quad (3.19)$$

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (\mathbf{x}_i - \mu_B)^2 \quad (3.20)$$

gdje B označava mini grupu, a x vrijednost ulaza. Nelinearnost se onda primjenjuje na tako normaliziranom izlazu.

4. Podatkovni skup za učenje

Kako je već rečeno, u ovom radu učenje je nadzirano. Podatkovni skup KITTI jedan je od skupova koji se koriste u učenju postupka stereoskopske rekonstrukcije. U sljedećem pododjeljku bit će predstavljen podatkovni skup KITTI 2015, njegove specifičnosti, te postupci pripreme podataka.

4.1. Podatkovni skup KITTI 2015

Podatkovni skup KITTI 2015, skup je od 200 stereo parova slika za koje su laserski izmjerene mape dispariteta. Stereo par slika je unaprijed kalibriran i rektificiran. Primjer uz odgovarajuće disparitete prikazan je na slici 4.1. Podatkovni skup je često korišten u postupcima stereoskopske rekonstrukcije za autonomnu vožnju. Podatkovni skup je podijeljen na skup za učenje kojeg čini 80% slika, sljedećih 20% čini skup za validaciju.

Mape dispariteta su vrlo rijetke, odnosno dispariteti su izmjereni laserski za manji podskup piksela. Takve mape će na mjestima piksela za koje nije izmjeren disparitet imati vrijednost nula. Primjer takvih piksela su oni koji prikazuju udaljene objekte, kao što je nebo. Takvo mjerenje proizlazi iz toga što je disparitet obrnuto proporcionalan udaljenosti, te poprima vrijednost nula u beskonačnosti.

Također napravljena je intervencija na područjima slike na kojima se nalazi automobil. Intervencija se sastojala od proglašavanja dispariteta na mjestima automobila. Ova intervencija je od velike važnosti rekonstrukcijskim algoritmima jer se na automobilu nalaze refleksivne površine poput stakla koje u konkurentnim stereo kamerama mogu izgledati drugačije.[8]



Slika 4.1: Primjer iz podatkovnog skupa KITTI 2015. Gornja slika prikazuje referentnu lijevu sliku, a donja slika prikazuje disparitet za odgovarajuću sliku.

5. Duboko učenje u kontekstu stereoskopske rekonstrukcije

U ovom poglavlju bit će predstavljena arhitektura dubokog modela koji se koristi za stereoskopsku rekonstrukciju. Metoda koristi ugrađivanje okana u visokodimenzionalan metrički prostor, pri čemu je moguće uspoređivanje po sličnosti koje koristimo u traženju korespondentnih piksela.

5.1. Ugrađivanje okana u metrički prostor

Stereo algoritmi zahtijevaju robusne metrike s obzirom na šum, dok s druge strane tražimo da metrika na pikselima bez teksture nosi određenu informaciju. U ovom radu razmatran duboki model koji uči na slikama s poznatim disparitetima zadovoljava uvjete stereo algoritama.

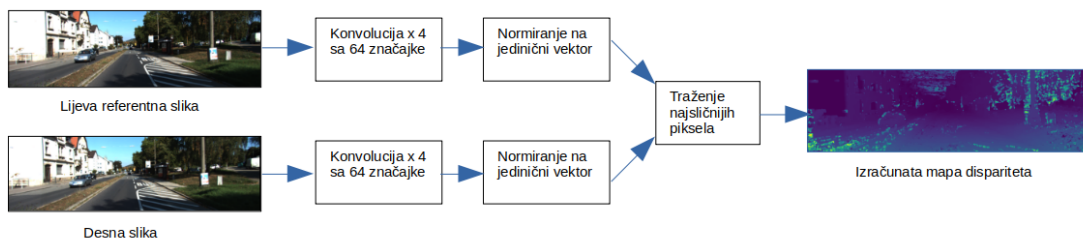
U ovom radu se koristi duboki model za ugrađivanje slika u visokodimenzionalan prostor.[7] Duboki model sastoji se od 4 konvolucijska sloja, pri čemu svaki sloj ima 64 značajke (engl. features). Za konvoluciju je korištena jezgra veličine 3×3 . Na ulaz modela dovodimo slikovno okno koje je dimenzija $P \times P$, dok je izlaz modela ugrađivanje okna u 64 dimenzionalan vektor. Ono što konvolucijskoj arhitekturi daje prednost je što je moguće ugrađivanje provesti u jednom prolazu nad cijelom slikom. Prilikom takvog ugrađivanja koristi se nadopunjavanje nulama da se dobije izlaz istih prvih dviju dimenzija, nadalje ulaz je slika dimenzija $W \times H$, a izlaz ugrađivanje čije su dimenzije $W \times H \times 64$. Kako bismo mogli koristiti kosinusnu sličnost (objašnjeno kasnije), potrebno je normirati izlaze na jedinični vektor duž osi značajki.

Nakon što je dubokim modelom ostvareno ugrađivanje okana u visokodimenzionalan prostor, moguće je provesti izračun mape dispariteta. Razmatramo jedan par slika I_L i I_D . Provedenim ugrađivanjem nastaju E_L i E_D spomenutih dimenzija $W \times H \times 64$. Nad takvim reprezentacijama tražimo korespondentne piksele. Kako je već spomenuto slike su rektificirane, što nam omogućuje usporedbu samo po horizontalnoj ulici. U ovom radu razmatrat će se dispariteti u intervalu $[0 \dots D]$, pri čemu je D maksimalan disparitet u skupu KITTI i iznosi 229. Kako je lijeva slika referentna, desna ugrađivanja posmičemo za disparitete u razmatranom intervalu uzduž horizontalne osi. Nakon posmaka vrši se skalarni umnožak vektora na odgovarajućim indeksima. Takvim množenjem dobiva se vektor dimenzija $W \times H \times D$, na čijim trećim dimenzijama se nalaze mjere sličnosti piksela.

Opisana mjera sličnosti vektora naziva se *kosinusna sličnost vektora*. U nastavku je dan matematički izraz takve mjere sličnosti:

$$\frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} = \frac{\sum_i a_i b_i}{(\sum_i a_i^2)(\sum_i b_i^2)} \quad (5.1)$$

Kako je cilj pronaći najbližije piksele koristi se operacija *argmax* po trećoj dimenziji. Funkcija *argmax* će vratiti indeks najbližijeg piksela, odnosno čija je mjera sličnosti najveća. Provedenim operacijama je nastala tzv. mapa dispariteta. Kako se u postupku ne uzima u obzir susjedstvo piksela, ovakav pristup se naziva i *Winner takes all* zato što se razmatraju samo najbližiji pikseli. Na slici 5.1. prikazan je postupak izračunavanja mape dispariteta.



Slika 5.1: Postupak računanja mape dispariteta. Stereo par slika iz konkurentnih kamera ide na ulaz, prolazi kroz 4 kovolucijska sloja, te se na kraju izlaz normira na jedinični vektor. U zadnjem koraku se pronalaze nasličniji pikseli metodom *Winner takes all*.

5.2. Učenje modela za ostvarivanje korespodencije

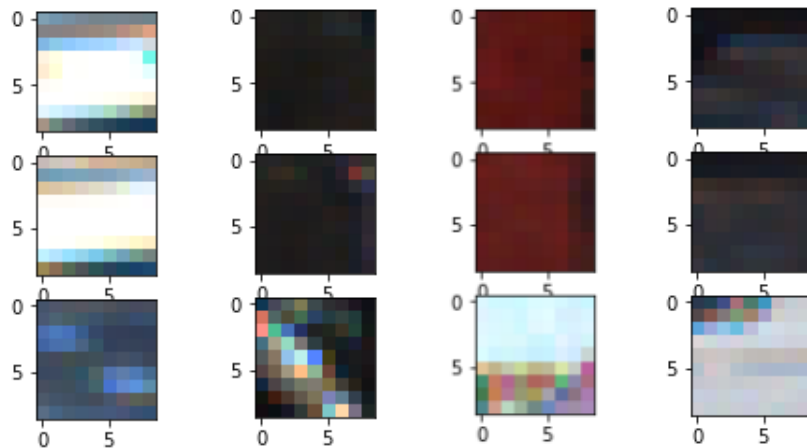
Prije no što se krene s učenjem potrebno je svesti piksele slika na normalnu jediničnu razdiobu. Za takvo što je potrebno izračunati srednju vrijednost μ i standardna devijacija σ , te je konačan izraz za piksele $I(x, y)$:

$$I'(x, y) = \frac{I(x, y) - \mu}{\sigma} \quad (5.2)$$

Nakon što je izvršena normalizacija izdvajaju se okna nad kojima će spomenuti duboki model učiti. Ukupan skup sastoji se od oko 16 milijuna primjera, a za svaki od njih poznat je disparitet. Primjer za učenje sastoji od tri okna, a to su referentno okno R, pozitivno okno P i negativno okno N. Referentno i pozitivno okno čine međusobno sličan par, dok referentno i negativno čine manje sličan par. U nastavku je objašnjena ekstrakcija okna:

- $R = I_L^{PxP}(x, y)$ je okno lijeve slike i predstavlja referentno okno
- $P = I_D^{PxP}(x - d, y)$ je pozitivno okno, a dobiva se horizontalnim posmakom središta za poznati disparitet d
- $P = I_D^{PxP}(x - d - o, y)$ je negativno okno, a dobiva se horizontalnim posmakom središta za disparitet d te dodatnim nasumičnim odmakom o . Odmak se odabire iz intervala $[-14, -4] \cup [4, 14]$.

Na slici 6.2. prikazano je nekoliko primjera za učenje.



Slika 5.2: Prikaz 4 primjera za učenje. Prvi redak predstavlja referentno lijevo okno, redak ispod prikazuje čvrsti pozitiv (pozitivno okno bez nasumičnog pomaka), te zadnji redak prikazuje negativno okno.

Tijekom postupka optimizacije minimizira se funkcija gubitka čiji je izraz dan u nastavku:

$$L(R, P, N) = \max(0, m + \mathbf{h}(\mathbf{R}|\theta) \cdot \mathbf{h}(\mathbf{N}|\theta) - \mathbf{h}(\mathbf{R}|\theta) \cdot \mathbf{h}(\mathbf{P}|\theta)) \quad (5.3)$$

pri čemu h predstavlja izlaz modela s parametrima θ . U funkciji gubitka zahtijevano je da je ugrađivanje referentnog okna sličnije ugrađivanju pozitivnog okna u odnosu na ugrađivanje negativnog okna za najmanje m . Pojednostavljeno rečeno, model će učiti samo na primjerima na kojima je sličnost R i P manja od sličnosti između R i N za više od m . [10]

6. Eksperimentalni rezultati

Kvalitetu rekonstrukcijske točnosti ocjenjuje se na slikama za koje su poznati dispariteti. Svi eksperimenti provedeni su na predstavljenom skupu KITTI. Za točno rekonstruirani piksel smatrat će se ako je odstupanje od točnog manje od 3. U tu svrhu koristit će se skup za testiranje koji se sastoji od 20% slika početnog skupa od 200 slika.

Učenje traje 14 epoha, algoritam učenja je ADAM uz stopu učenja koja iznosi $1e^{-3}$ do 10. epohe, a nakon nje je $1e^{-4}$. Veličina mini-grupe (engl. batch-size) je 128. Svi eksperimenti izvodili su se na platformi Colab pro, a učenje je trajalo 9 sati.

Provedena su sljedeća 3 eksperimenta:

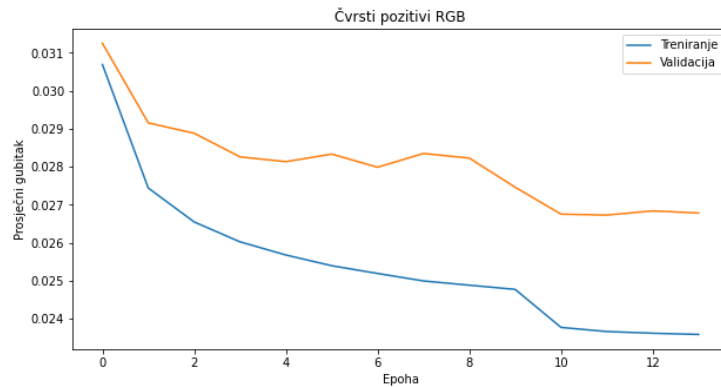
- Ulazne slike u boji (RGB) s čvrstim pozitivima, odnosno bez posmaka
- Ulazne slike u boji (RGB) s rastresenim pozitivima, odnosno nasumično posmaknutim brojem iz skupa $\{-1, 0, 1\}$
- Sive ulazne slike s čvrstim pozitivima

Tablica 6.1: Rekonstrukcijske točnosti razmatrani modela

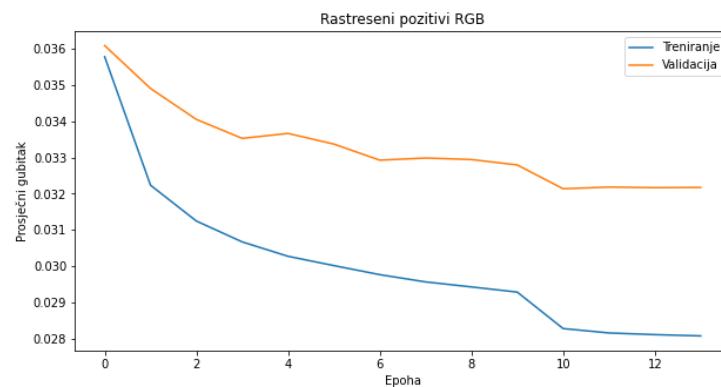
Model	Treniranje	Testiranje
Ulazne slike u boji (RGB) s čvrstim pozitivima	82.90%	81.24%
Ulazne slike u boji (RGB) s rastresenim pozitivima	82.43%	80.81%
Sive ulazne slike s čvrstim pozitivima		

U tablici 6.1. prikazane su rekonstrukcijske točnosti razmatranih eksperimenata. Najbolji od 3 razmatrana modela je onaj koji koristi ulazne slike u boji s čvrstim pozitivima.

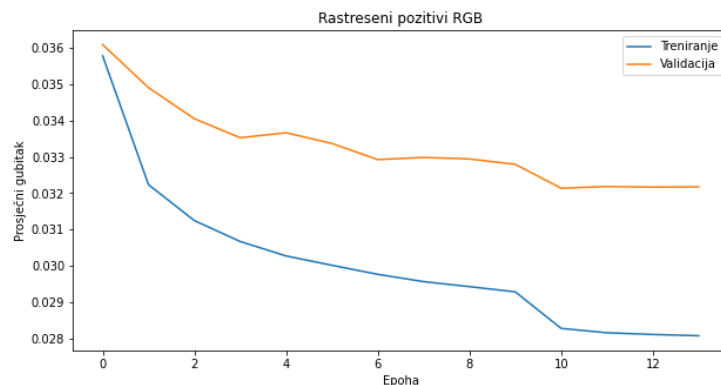
Na slikama 6.1, 6.2 i 6.3 prikazano je kretanje prosječnih gubitaka po epohama i to na skupovima za treniranje i validaciju. Slike prikazuju grafove za model s čvrstim i rastresenim pozitivima za RGB, te model s čvrstim pozitivima za sive ulazne slike. Vidljivo je kako model koji koristi čvrste pozitive brže minimizira pogrešku. Što se tiče sivih slika s čvrstim pozitivima u odnosu na RGB slike s čvrstim pozitivima, model s RGB slikama je bolji, razlog leži u tome što RGB slike imaju 3 kanala, te posljedično nose više informacija.



Slika 6.1: Graf prikazuje kretanje prosječnog gubitka tijekom epoha treniranja i validacije. Za učenje modela se koriste čvrsti pozitivi RGB slika, odnosno bez posmaka.



Slika 6.2: Graf prikazuje kretanje prosječnog gubitka tijekom epoha treniranja i validacije. Za učenje modela se koriste rastreseni pozitivi RGB slika (posmaknuti za nasumičan broj u skupu $\{-1, 0, 1\}$).



Slika 6.3: Graf prikazuje kretanje prosječnog gubitka tijekom epoha treniranja i validacije. Za učenje modela se koriste čvrsti pozitivni sivih slika.

Na slici 6.4. prikazane su rekonstrukcije za 4 slike nasumično odabrane iz skupa. Na mapama dispariteta je vidljiva velika razlika u susjednim pikselima, što je posljedica nekorištenja nekog od filtera zaglađivanja. Razmatrana metoda griješi na mjestima slike na kojima se nalaze reflektivne površine automobila. Na takvim područjima se svjetlost odbija, te okna iz tih područja predstavljaju problem modelu u procesu učenja. Sljedeća pojava koja predstavlja problem modelu su područja koja se na slikama jedne kamere vide, a na slikama druge kamere ne. Ta pojava naziva se stereoskopskom sjenom.[5]



Slika 6.4: Prikaz rekonstrukcijske točnosti modela za ugrađivanje okana u visokodimenzionalan prostor. Lijeve slike prikazuju slike iz lijeve referentne kamere. U sredini se nalaze procjenjene mape dispariteta. Na mapi dispariteta što je svjetliji piksel to je veći disparitet, a tamniji što je manji. Zdesna se nalaze rekonstrukcijske točnosti, pri čemu su crnom bojom označeni pikseli uz nepoznat disparitet, plavom bojom pogrešno rekonstruirani pikseli, te zelenom bojom točno rekonstruirani pikseli.

7. Programska izvedba i vanjske biblioteke

Za izradu ovog rada korišten je programski jezik Python. Za izgradnju i učenje modela korišteno je radno okruženje PyTorch. Ovo okruženje nudi velik spektar alata i podrške, a jedan od njih je automatska diferencijacija, što uvelike ubrzava proces izgradnje i učenja modela. Torch kao sastavna jedinica PyTorch-a omogućava izračune tenzorima na velikoj paleti grafičkih kartica kompanije Nvidia.

Za pripremu podataka i računanje tenzorima korištena je biblioteka Numpy. Ova biblioteka pisana je u jeziku niske razine C, što čini operacije vrlo brzim.

Za proces učenja, validacije i testiranja korištena je platforma Colab Pro u kombinaciji s Google Drive. Platforma u pretplati Pro nudi 3 grafičke kartice koje se dodjeljuju ovisno o raspoloživosti. Tipovi grafičkih kartica su sljedeći: K80, T4 i P100.

Podatkovni skup KITTI 2015 uzet je s javno dostupne stranice u vlasništvu kolaboracije Karlsruhe Institute of Technology i Toyota Technological Institute.

8. Zaključak

LITERATURA

- [1] Akash Ajagekar. Adam. URL <https://optimization.cbe.cornell.edu/index.php?title=Adam>.
- [2] Nikola Bunjevac. Gusta stereoskopska rekonstrukcija poluglobalnim podudaranjem. 2017. URL <http://www.zemris.fer.hr/~ssegvic/project/pubs/bunjevac17bs.pdf>.
- [3] Richard Szeliski Daniel Scharstein. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. URL <https://vision.middlebury.edu/stereo/taxonomy-IJCV.pdf>.
- [4] Siniša Šegvić. Neslužbene stranice predmeta duboko učenje 1. URL <http://www.zemris.fer.hr/~ssegvic/du/>.
- [5] Antonio Ilinović. Učenje korespondencijskog ugrađivanja za stereoskopsku rekonstrukciju. 2022. URL <http://www.zemris.fer.hr/~ssegvic/project/pubs/ilinovic22bs.pdf>.
- [6] Alfred Wong Jonathon Price. Stochastic gradient descent. URL https://optimization.cbe.cornell.edu/index.php?title=Stochastic_gradient_descent.
- [7] Yann LeCun Jure Žbontar. Computing the stereo matching cost with a convolutional neural network. 2015. URL <https://ieeexplore.ieee.org/document/7298767/>.
- [8] Yann LeCun Jure Žbontar. Stereo matching by training a convolutional neural network to compare image patches. 2016. URL <https://arxiv.org/abs/1510.05970/>.
- [9] Dino Kovač. Gusta stereoskopska rekonstrukcija scene predstavljene ravninskim

odsječcima. 2015. URL <http://www.zemris.fer.hr/~ssegvic/project/pubs/kovac15ms.pdf>.

- [10] Marin Oršić. Učenje korespondencijske metrike za gustu stereoskopsku rekonstrukciju. 2017. URL <http://www.zemris.fer.hr/~ssegvic/project/pubs/orsic17ms.pdf>.

Korespondencijska ugrađivanja za stereoskopsku rekonstrukciju

Sažetak

Sažetak na hrvatskom jeziku.

Ključne riječi: Ključne riječi, odvojene zarezima.

Title

Abstract

Abstract.

Keywords: Keywords.