SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1268

MODELIRANJE SUPOJAVLJIVANJA SEMANTIČKIH OZNAKA UVJETNIM SLUČAJNIM POLJIMA

Denis Čaušević

Zagreb, lipanj 2016.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
ODBOR ZA DIPLOMSKI RAD PROFILA

Zagreb, 11. ožujka 2016.

Predmet: **Diplomski rad**

DIPLOMSKI ZADATAK br. 1268

Pristupnik: **Denis Čaušević (0036462803)**
Studij: Računarstvo
Profil: Računarska znanost

Zadatak: **Modeliranje supojavljivanja semantičkih oznaka uvjetnim slučajnim poljima**

Opis zadatka:

Semantička segmentacija prirodnih scena jest preslikavanje slikovnih elemenata zadane slike u unaprijed zadani skup semantičkih oznaka. Većina postojećih metoda ovaj zadatak rješava pod pretpostavkom međusobne nezavisnosti slikovnih elemenata. Međutim, to u praksi najčešće nije slučaj. Ako smo u nekom elementu slike prepoznali automobil, sva je prilika da će u njegovom susjedstvu biti još elemenata tog razreda. Zbog toga je ovaj rad posvećen metodama koje konačan rezultat segmentacije određuju globalnom optimizacijom polja semantičkih oznaka. Posebno su nam zanimljive kriterijske funkcije koje kombiniraju i) nezavisnu izglednost razreda s obzirom na lokalno susjedstvo slikovnog elementa i ii) izglednost supojavljivanja bliskih slikovnih elemenata s obzirom na njihov razred, boju i udaljenost.
U okviru rada, potrebno je iz literature proučiti načine modeliranja unarnih i binarnih izglednosti semantičkih razreda kao i ostvarivanje interakcije tih potencijala uvjetnim slučajnim poljima. Konstruirati binaran potencijal slikovnih elemenata definiran razredima elemenata, njihovom međusobnom udaljenošću te sličnošću njihovih boja. Uhodati zaključivanje korištenjem aproksimacije srednjeg polja te postupak za validiranje hiperparametara uvjetnog slučajnog polja. Vrednovati razvijene postupke u kombinaciji s unarnim potencijalima dobivenima dubokim neuronskim mrežama na skupovima podataka KITTI i Cityscapes. Prikazati i ocijeniti ostvarene rezultate. Predložiti pravce budućeg razvoja.

Zadatak uručen pristupniku: 18. ožujka 2016.
Rok za predaju rada: 1. srpnja 2016.

Mentor:

Izv. prof. dr. sc. Siniša Šegvić

Predsjednik odbora za diplomski rad profila:

Prof. dr. sc. Siniša Srbljić

Djelovođa:

Doc. dr. sc. Tomislav Hrkać

Zahvaljujem se mentoru, izv. prof. dr. sc. Siniši Šegviću na kontinuiranoj podršci i vodstvu kroz diplomski studij, a posebice na savjetima i pomoći prilikom izrade ovog rada.

Zahvaljujem se i mag. ing. Ivanu Kreši na velikoj pomoći prilikom implementacije modela opisanog u radu, te dr. sc. Josipu Krapcu na pruženim savjetima i idejama tijekom izrade rada.

Sadržaj

1. Uvod

Semantičko razumijevanje scene jedan je od glavnih ciljeva računalnog vida. Problemu semantičkog razumijevanja može se pristupiti semantičkom segmentacijom, koja za cilj ima svakom pikselu izvorne slike pridružiti oznaku semantičkog razreda kojem navedeni piksel pripada. Primjeri semantičkih razreda koji se javljaju u urbanim prometnim scenama uključuju pješake, zgrade, automobile i sl. Sustavi za semantičku segmentaciju trebaju svaki piksel slike, koja prikazuje primjerice prometnu scenu, klasificirati u jedan od prethodno spomenutih razreda.

Semantička segmentacija primjenjuje se u raznim područjima ljudske djelatnosti. Najzanimljivije je područje autonomnih vozila, gdje se segmentirana slika koristi za razumijevanje scene, te pomaže u praćenju ceste, pješaka i vozila s ciljem izbjegavanja nesreća.

Većina problema računalnog vida tipično se rješava primjenom modela strojnog učenja koji opisuju odnos između slikovnih značajki i informacije visoke razine o sadržaju slike. Početni pristupi semantičkoj segmentaciji temeljili su se upravo na takvim modelima. Korištene su duboke konvolucijske neuronske mreže, koje su u stanju generirati kvalitetne slikovne značajke. Generirane značajke potom su korištene kako bi se pikseli klasificirali u pripadne semantičke razrede. Nedostatak opisanog postupka je izostanak modeliranja interakcije među informacijama visoke razine. Cilj je modelirati sljedeću i slične vrste interakcija: ako je piksel na poziciji (i, j) s velikom sigurnošću klasificiran u razred „automobil“, onda je velika vjerojatnost da i susjedni piksel $(i + 1, j)$, ukoliko je dovoljno sličan prethodnom, pripada istom razredu. Navedene interakcije mogu se modelirati uvjetnim slučajnim poljima.

Uvjetna slučajna polja su napredni model strojnog učenja, često korišten za modeliranje interakcije između prostorno bliskih slikovnih elemenata. U početku su uvjetna slučajna polja bila ograničena na modeliranje relacija samo između neposredno susjednih piksela zbog složenosti zaključivanja. Međutim, primjena metode srednjeg polja omogućila je izvedbu aproksimativnog zaključivanja i za veća susjedstva. Modeliranje supojavljivanja semantičkih oznaka Gaussovim jezgrama definiranim nad vektorom boje i pozicijama piksela omogućilo je efikasnu izvedbu zaključivanja za potpuno povezana uvjetna slučajna polja s unarnim i binarnim

potencijalima. Takva uvjetna slučajna polja čine potpuno povezani graf nad varijablama koje predstavljaju semantičke oznake razreda piksela. Pojava spomenutih uvjetnih slučajnih polja omogućila je modeliranje složenijih interakcija među informacijama visoke razine, što je dovelo do pojave modela koji ostvaruju visoke performanse pri rješavanju problema semantičke segmentacije.

U ovom radu implementiran je sustav za semantičku segmentaciju, koji koristi konvolucijske neuronske mreže, uvjetno slučajno polje s ograničenim susjedstvom, te potpuno povezano uvjetno slučajno polje. Konvolucijska mreža koristi se za učenje unarnih i binarnih potencijala. Uvjetno slučajno polje kombinira unarne i binarne potencijale kako bi se dobila kvalitetnija predikcija u odnosu na unarne potencijale. Na kraju se izlazi uvjetnog slučajnog polja koriste za definiranje unarnih potencijala gusto povezanog uvjetnog slučajnog polja definiranog nad pikselima, koje dodatno poboljšava konačnu performansu modela.

U poglavlju 2 dan je kratki uvod u teoriju vjerojatnosti i probabilističke grafičke modele. Metoda srednjeg polja opisana je u poglavlju 3, dok poglavlje 4 sadrži kratki pregled neuronskih mreža i postupka njihovog učenja. Detalji implementiranog modela opisani su u poglavlju 5, a dobiveni eksperimentalni rezultati u poglavlju 6.

2. Probabilistički grafički modeli

U ovom poglavlju najprije se ukratko opisuju osnovni pojmovi iz teorije vjerojatnosti potrebni za razumijevanje probabilističkih grafičkih modela, prvenstveno uvjetnih slučajnih polja. Nakon toga, izlažu se osnove grafičkih modela s naglaskom na uvjetna slučajna polja, te se objašnjavaju osnove njihove primjene na probleme semantičke segmentacije.

2.1. Teorija vjerojatnosti

U teoriji vjerojatnosti razmatraju se događaji koji se mogu, ali ne moraju dogoditi. Pojam događaja najjednostavnije je opisati pomoću stohastičkog pokusa. Tako se naziva svaki pokus čiji ishod nije unaprijed određen [2]. Ishod takvog pokusa naziva se elementarni događaj. Bacanje kocke predstavlja jedan stohastički pokus, dok konačna strana na koju kocka padne predstavlja jedan elementarni događaj.

Skup svih elementarnih događaja označavamo s Ω i nazivamo prostorom događaja. Događajem nazivamo podskup od Ω, dakle skup elementarnih događaja. Događaje od interesa označavamo velikim slovima abecede, A, B, C ... Označimo sa S skup mjerljivih događaja, dakle događaja kojima želimo pridružiti vjerojatnosti. Događaj koji se uvijek ostvaruje nazivamo sigurnim događajem, dok događaj koji se pri realizaciji pokusa nikada neće ostvariti nazivamo nemoguć događaj i označavamo s $\emptyset$.

Teorija vjerojatnosti zahtijeva da prostor događaja zadovoljava sljedeća svojstva:

- Sadrži nemoguć i siguran događaj
- Zatvoren je s obzirom na operaciju unije
- Zatvoren je s obzirom na operaciju komplementa

Vjerojatnosna distribucija definirana nad (Ω, S) je preslikavanje $P: S \rightarrow [0,1]$ koje zadovoljava sljedeće uvjete:

- $\forall A \in S,\ P(A) \geq 0$
- $P(\Omega) = 1, P(\emptyset) = 0$
- $Ako\ je\ A, B \in S\ i\ A \cap B = \emptyset$ onda vrijedi $P(A \cup B) = P(A) + P(B)$ (1)

Prva dva uvjeta ograničavaju vrijednost vjerojatnosti, dok treći definira da je vjerojatnost unije međusobno isključivih događaja jednaka zbroju vjerojatnosti pojedinačnih događaja.

Uz pojmove događaja i vjerojatnosti vezane su i slučajne varijable. Slučajna varijabla je funkcija $X: \Omega \rightarrow Val(X)$ definirana na skupu elementarnih događaja Ω. $Val(X)$ predstavlja skup mogućih realizacija slučajne varijable X. Ovisno o tome poprima li slučajna varijabla samo diskretne ili kontinuirane vrijednosti, razlikujemo diskretne i kontinuirane slučajne varijable. Slučajna varijabla predstavlja način opisa pojedinog svojstva ishoda stohastičkog pokusa. Neka je zadana slučajna varijabla X, koja opisuje broj dobiven bacanjem kocke. Vrijednosti koje ta varijabla može poprimiti su $Val(X) \in \{1,2,3,4,5,6\}$. Pojedini iskaz o slučajnoj varijabli, poput $X = 5$, predstavlja jedan događaj (u ovom slučaju događaj da je kocka pala na stranu s brojem 5). Slučajna varijabla X poprima vrijednosti iz skupa $Val(X)$ s određenom vjerojatnosti. Vjerojatnost je mjera koja zadovoljava aksiome (1).

U nastavku ovog rada koristi se sljedeći dogovor: slučajne varijable označavaju se velikim slovima abecede $(A, B, C \dots)$, a njihove realizacije ekvivalentnim malim slovima abecede $(a, b, c \dots)$. Vjerojatnost da slučajna varijabla X poprimi vrijednost x označava se kao $P(X = x)$ ili skraćeno $P(x)$. Distribucija slučajne varijable X označava se kao $P(X)$.

Uvjetna vjerojatnost događaja A uz poznatu realizaciju događaja B definirana je na sljedeći način:

$$P(A = a|B = b) = \frac{P(A = a, B = b)}{P(B = b)} \quad (2)$$

Iz definicije uvjetne vjerojatnosti (2) vidljivo je da vrijedi sljedeće:

$$P(A = a, B = b) = P(B = b)P(A = a|B = b) \quad (3)$$

Prethodni izraz (3) poznat je kao pravilo produkta, te se njegovim poopćenjem dobiva lančano pravilo uvjetnih vjerojatnosti:

$$P(x_1, x_2, \dots, x_k) = \prod_{i=1}^{k} P(x_i|x_1, x_2, \dots, x_{k-1}) \quad (4)$$

Lančano pravilo je bitno budući da omogućava faktorizaciju zajedničke vjerojatnosti $P(x_1, x_2, \dots, x_k)$, što je osnova probabilističkih grafičkih modela.

Aposteriorne vjerojatnosti događaja A i B povezane su preko Bayesovog pravila (5).

$$P(A = a|B = b) = \frac{P(B = b|A = a)P(A = a)}{P(B = b)} \tag{5}$$

Svi prethodno navedeni izrazi mogu se promatrati u kontekstu slučajnih varijabli i njihovih distribucija. Slučajne varijable se najčešće ne promatraju u izolaciji, nego nas istovremeno zanima ishod više slučajnih varijabli. U tom slučaju razmatra se njihova zajednička distribucija $P(X_1, X_2, \dots, X_n)$, koja svakoj kombinaciji njihovih vrijednosti pridružuje određenu vjerojatnost. Na temelju zajedničke vjerojatnosti moguće je izračunati marginalnu vjerojatnost pojedine slučajne varijable pravilom zbroja:

$$P(x_i) = \sum_{x_1, x_2, .., x_{i-1}, x_{i+1}, .., x_n} P(x_1, x_2, \dots, x_n) \tag{6}$$

Za definiranje probabilističkih grafičkih modela, bitni su pojmovi nezavisnosti i uvjetne nezavisnosti slučajnih varijabli budući da omogućavaju pojednostavljenje faktorizacije zajedničke vjerojatnosti definirane lančanim pravilom (4).

Za dvije slučajne varijable X i Y kažemo da su nezavisne ako poznavanje realizacije jedne varijable nema nikakvog utjecaja na vjerojatnost druge:

$$\begin{gathered} P(x, y) = P(x)P(y) \\ P(x|y) = P(x) \qquad P(y|x) = P(y) \end{gathered} \tag{7}$$

Slučajne varijable X i Y su uvjetno nezavisne uz danu varijablu Z, što označavamo kao $X \perp Y \mid Z$, ako i samo ako vrijedi:

$$P(x|y, z) = P(x|z) \tag{8}$$

Ili, ekvivalentno:

$$P(x, y|z) = P(x|z)P(y|z) \tag{9}$$

Intuitivno, varijable X i Y su uvjetno nezavisne ako, jednom kada je poznat ishod varijable Z, znanje o ishodu varijable Y ni na koji način ne utječe na ishod varijable X (i obrnuto) [3].

Za potrebe ovog diplomskog rada također je bitan i pojam očekivanja. Neka je X diskretna slučajna varijabla koja poprima numeričke vrijednosti. Onda je očekivanje od X pod distribucijom P definirano kao:

$$E_P[X] = \sum_x xP(x) \tag{10}$$

Očekivanje ima svojstvo linearnosti:

$$E[c_1X + c_2Y] = c_1E[X] + c_2E[Y] \tag{11}$$

Ukoliko su slučajne varijable X i Y nezavisne, vrijedi:

$$E[XY] = E[X]E[Y] \tag{12}$$

Uvjetno očekivanje slučajne varijable X uz zadani $\boldsymbol{y}$ (dokaz, promatrana realizacija nekog događaja) definirano je kao:

$$E_P[X|\boldsymbol{y}] = \sum_x xP(x|\boldsymbol{y}) \tag{13}$$

2.2. Grafički modeli

Grafički modeli predstavljaju moćan alat za predstavljanje i zaključivanje temeljeno na vjerojatnosnim distribucijama. Naivni pristup reprezentaciji vjerojatnosne distribucije velikog broja slučajnih varijabli je prostorno veoma skup. Primjerice, naivna reprezentacija zajedničke distribucije n binarnih varijabli, zahtijevala bi pohranjivanje $2^n - 1$ realnih brojeva. Međutim, distribucija velikog broja slučajnih varijabli najčešće se može prikazati kao produkt lokalnih funkcija, tj. faktora, koji ovise o manjim podskupovima varijabli. Opisana faktorizacija, usko povezana s relacijama uvjetne nezavisnosti između varijabli, pogodna je za reprezentaciju pomoću grafova. Takva reprezentacija istovremeno pruža mogućnost primjene algoritama razvijenih za grafove na problem zaključivanja kod probabilističkih grafičkih modela.

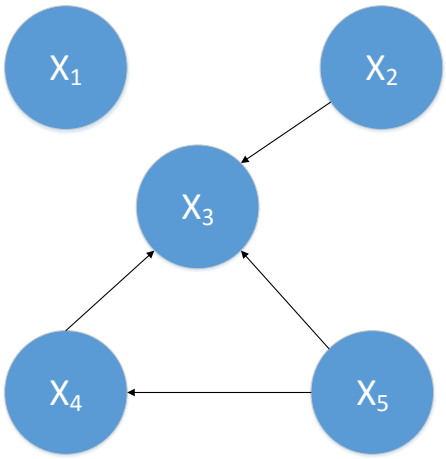
2.2.1. Usmjereni grafički modeli

Usmjereni grafički model opisuje kako je vjerojatnosna distribucija faktorizirana na lokalne uvjetne distribucije. Neka je G usmjereni aciklički graf, čiji su vrhovi slučajne varijable $X_1, X_2, \dots, X_n$, te neka su $\pi(s)$ indeksi svih roditelja čvora X_s u grafu G. Usmjereni grafički model je familija distribucija faktorizirana kao:

$$P(\boldsymbol{X}) = \prod_{s=1}^{n} P(X_s|\boldsymbol{X}_{\pi(s)}) \tag{14}$$

U slučaju da je $\pi(s)$ prazan skup, pripadna distribucija se promatra kao $P(X_s)$. Ovako definirani grafički modeli nazivaju se još i Bayesovim mrežama. Primjer

usmjerenog grafičkog modela prikazan je na slici 1, dok je pripadna faktorizacija vjerojatnosne distribucije predstavljena izrazom (28).



Slika 1 Primjer Bayesove mreže

$$P(X_1, X_2, \dots, X_5) = P(X_1)P(X_2)P(X_3|X_2, X_4, X_5)P(X_4|X_5)P(X_5) \tag{15}$$

Bridovi u grafu opisuju nezavisnosti između varijabli. Ako su varijable zavisne između njih postoji usmjereni brid, u suprotnom ne.

2.2.2. Neusmjereni grafički modeli

Pretpostavimo da se vjerojatnosna distribucija može prikazati kao produkt faktora $\phi(\boldsymbol{X}_\phi)$, gdje $\boldsymbol{X}_\phi$ predstavlja podskup varijabli nad kojima je definiran faktor ϕ. Skup svih faktora označavamo sa Φ. Pritom vrijednost svakog faktora je nenegativna, te predstavlja stupanj slaganja vrijednosti $\boldsymbol{x}_\phi$ jednih s drugima. Pridruživanje vrijednosti $\boldsymbol{x}_\phi$ varijablama $\boldsymbol{X}_\phi$ za koje je vrijednost faktora $\phi(\boldsymbol{x}_\phi)$ odgovara većoj vjerojatnosti.

Neusmjereni grafički model je familija vjerojatnosnih distribucija koje se faktoriziraju u skladu s zadanim skupom faktora Φ. Za zadani skup faktora Φ, neusmjereni grafički model je skup distribucija koje se mogu zapisati kao

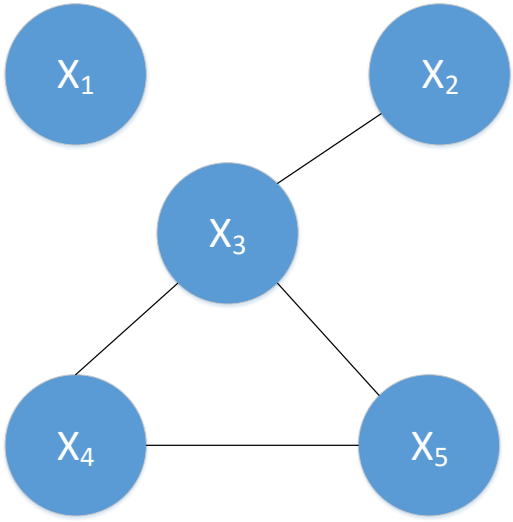
$$P(\boldsymbol{X}) = \frac{1}{Z}\prod_{\phi\in\Phi}\phi(\boldsymbol{X}_\phi) \tag{16}$$

Faktori ϕ pritom moraju zadovoljavati uvjet $\phi(\boldsymbol{x}_\phi) \geq 0$ za svaki $\boldsymbol{x}_\phi$. Faktori nisu definirani nad svim varijablama $\boldsymbol{X}$, nego nad podskupovima $\boldsymbol{X}_\phi \subset \boldsymbol{X}$, koji se još nazivaju i klikama (*engl. cliques*). Z je particijska funkcija i predstavlja normalizacijski član koji osigurava da konačni izraz bude vjerojatnosna distribucija:

$$Z = \sum_{\boldsymbol{x}} \prod_{\phi \in \Phi} \phi(\boldsymbol{x}_\phi) \tag{17}$$

Izračun particijske funkcije (17) je računski veoma zahtijevan budući da uključuje zbroj po svim mogućim vrijednostima varijabli $\boldsymbol{X}$, te se u praksi nastoji izbjeći.

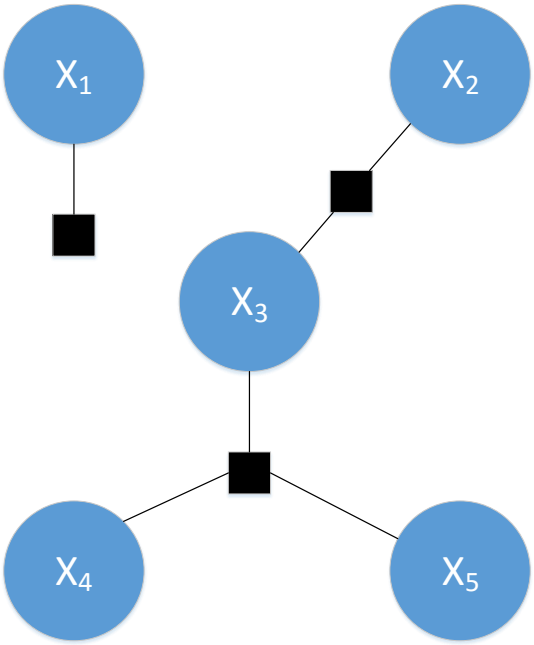
Neusmjereni grafički modeli prikazuju se pomoću neusmjerenih grafova. Vrhovi grafa G odgovaraju slučajnim varijablama $\boldsymbol{X}$, dok su bridovi definirani faktorima. Pritom za svaki faktor ϕ čvorovi $\boldsymbol{X}_\phi$ čine potpuno povezani podgraf u grafu G. Primjer neusmjerenog grafičkog modela koji opisuje faktorizaciju distribucije (31) prikazan je na slici 2.



Slika 2 Primjer neusmjerenog grafičkog modela

$$P(X_1, X_2, \dots, X_5) = \phi(X_1)\phi(X_2, X_3)\phi(X_3, X_4, X_5) \tag{18}$$

Navedeni grafički prikaz može ponekad biti nejednoznačan. Podgraf $\{X_3, X_4, X_5\}$ se može protumačiti kao jedan faktor definiran nad tri varijable, ili pak kao tri faktora definirana nad dvije varijable. Zbog toga se često umjesto običnih neusmjerenih grafova koriste faktorski grafovi koji jednoznačno prikazuju faktorizaciju vjerojatnosne distribucije. Faktorski graf je bipartitni graf $G = (V, F, E)$ u kojem skup čvorova V predstavlja slučajne varijable $\boldsymbol{X}$, dok drugi skup čvorova F predstavlja faktore Φ. Pritom ako je neki čvor $X_s \in V$ povezan s čvorom $\phi \in F$, onda je faktor ϕ definiran nad slučajnom varijablom X_s. Faktorski graf koji jednoznačno opisuje distribuciju (18) prikazan je na slici 3.



Slika 3 Primjer faktorskog grafa

Opisani modeli nazivaju se još i Markovljevim mrežama. Formalno, kažemo da se distribucija P sa faktorima $\phi \in \Phi$ faktorizira preko Markovljeve mreže G ako svaki faktor ϕ predstavlja potpuno povezani podgraf od G [1].

Izraz (16) se najčešće koristi u modificiranom obliku (19), koji uklanja ograničenje nenegativnosti faktora.

$$P(\boldsymbol{X}) = \frac{1}{Z} e^{-\sum_{\phi \in \Phi}(-\log \phi(\boldsymbol{X}_\phi))} = \frac{1}{Z} e^{-\sum_{\psi \in \Psi} \psi(\boldsymbol{X}_\psi)} \qquad (19)$$

Pritom su izvorni faktori ϕ zamijenjeni svojim negativnim logaritmima $\psi = -\log \phi$. Definirani potencijali ψ mogu poprimati pozitivne i negativne vrijednosti, te imaju drugačiju semantiku u odnosu na pripadne faktore ϕ. Vrijednost potencijala $\psi(\boldsymbol{x}_\psi)$ može se tumačiti kao cijena pridruživanja vrijednosti $\boldsymbol{x}_\psi$ slučajnim varijablama $\boldsymbol{X}_\psi$. Vjerojatnosna distribucija napisana u obliku (19) naziva se Gibbsovom distribucijom [3]. U statističkoj mehanici i termodinamici navedena distribucija poznata je kao Boltzmannova distribucija, te se koristi za opis vjerojatnosti određenog stanja fizikalnog sustava kao funkcije energije stanja i temperature sustava na koji se distribucija primjenjuje. U tom pogledu zbroj u izrazu (19) naziva se energijom, a proces učenja modela minimizacijom energije.

2.3. Uvjetna slučajna polja

U prethodnim poglavljima sve slučajne varijable $\boldsymbol{X}$ promatrane su u jednakom kontekstu. Međutim, u praksi su varijable podijeljene na skup ulaznih varijabli $\boldsymbol{X}$ čije su nam realizacije poznate (primjerice varijable koje modeliraju intenzitet piksela), te izlazne varijable $\boldsymbol{Y}$, koje želimo predvidjeti (primjerice semantički razredi piksela). Potrebno je izgraditi distribucije preko zajedničkog skupa varijabli $\boldsymbol{X} \cup \boldsymbol{Y}$. U tom se pogledu prethodna notacija može vrlo jednostavno proširiti varijablama $\boldsymbol{Y}$, što demonstrira proširenje izraza (19):

$$P(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{Z} e^{-\sum_{\psi \in \Psi} \psi(\boldsymbol{x}_\psi, \boldsymbol{Y}_\psi)} \tag{20}$$

Pritom je normalizacijska konstanta definirana kao:

$$Z = \sum_{x,y} e^{-\sum_{\psi \in \Psi} \psi(\boldsymbol{x}_\psi, \boldsymbol{y}_\psi)} \tag{21}$$

U praksi se najčešće koriste dvije vrste neusmjerenih grafičkih modela: Markovljeva i uvjetna slučajna polja.

Uvjetno slučajno polje modelira uvjetnu distribuciju $P(\boldsymbol{Y}|\boldsymbol{X})$, te se predstavlja kao neusmjereni graf G čiji su čvorovi $\boldsymbol{X} \cup \boldsymbol{Y}$. Neka neusmjereni graf G definira faktore $\phi \in \Phi$ takve da za svaki $\phi(\boldsymbol{D}_\phi)$ vrijedi $\boldsymbol{D}_\phi \not\subseteq \mathbf{X}$. Drugim riječima, niti jedan faktor nije definiran isključivo nad podskupom promatranih varijabli $\boldsymbol{X}$, nego uvijek sadrži barem jednu varijablu $Y_i \in \boldsymbol{Y}$. Tada neusmjereni graf definira uvjetnu distribuciju [1]:

$$\begin{aligned} P(\boldsymbol{Y}|\boldsymbol{X}) &= \frac{1}{Z(\boldsymbol{X})} \tilde{P}(\boldsymbol{Y}, \boldsymbol{X}) \\ \tilde{P}(\boldsymbol{Y}, \boldsymbol{X}) &= \prod_{\phi \in \Phi} \phi(\boldsymbol{D}_\phi) \\ Z(\boldsymbol{X}) &= \sum_{\boldsymbol{y}} \tilde{P}(\boldsymbol{Y}, \boldsymbol{X}) \end{aligned} \tag{22}$$

Dvije varijable u G su povezane neusmjerenim bridom ako se pojavljuju kao argumenti istog faktora [1]. Formalnije, uvjetno slučajno polje može se definirati na sljedeći način [5]:

Neka je $G = (V, E)$ graf definiran nad $\boldsymbol{Y} = (Y_v)_{v \in V}$, tako da je $\boldsymbol{Y}$ indeksiran vrhovima grafa G. Onda je $(\boldsymbol{X}, \boldsymbol{Y})$ uvjetno slučajno polje ako slučajne varijable Y_v zadovoljavaju

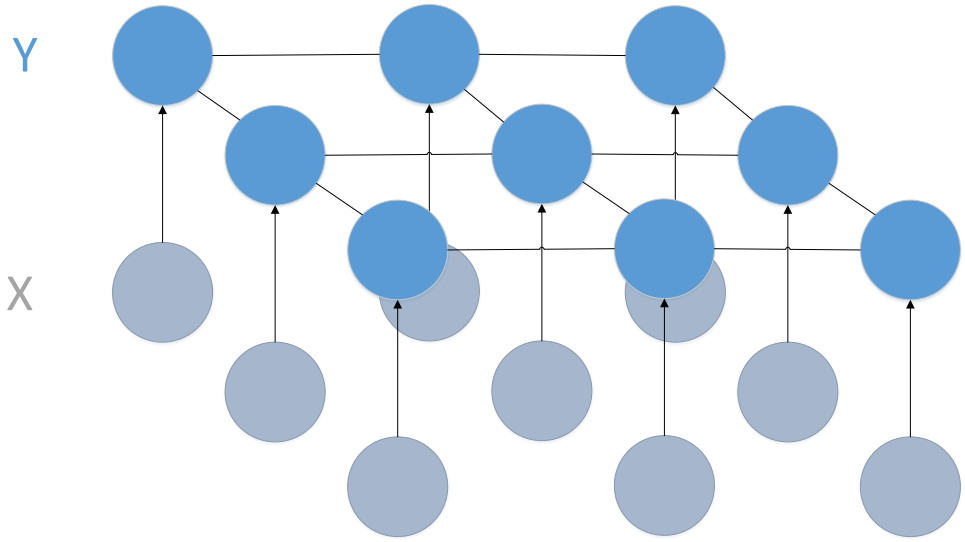
Markovljevo svojstvo s obzirom na graf kada su uvjetovane s $\boldsymbol{X}$: $P\left(Y_v \middle| \boldsymbol{X}, \boldsymbol{Y}_{w \in V \setminus \{v\}}\right) = P\left(Y_v \middle| \boldsymbol{X}, \boldsymbol{Y}_{w \in \{l \in V | l \sim v\}}\right)$, gdje $l \sim v$ označava da su čvorovi l i v susjedi u grafu G.

Markovljevo svojstvo osigurava da semantički razred y_v nekog piksela v ovisi samo o onim pikselima w s kojima je navedeni piksel neposredno povezan u grafu G. To je svojstvo bitno budući da olakšava zaključivanje u uvjetnim slučajnim poljima. Može se uočiti kako faktorizacija vjerojatnosne distribucije implicira Markovljevo svojstvo, dok obrat ne vrijedi [31]. Neka se $P(\boldsymbol{Y}|\boldsymbol{X})$ faktorizira u skladu s grafom $G = (V, E)$ kao $P(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{\phi_C \in \Phi} \phi_C\left(\boldsymbol{d}_{\phi_C}\right)$. Onda za sve disjunktne podskupove $A, B, S \subseteq V$ takve da je $A \perp B|S$ vrijedi $\boldsymbol{D}_A \perp \boldsymbol{D}_B|\boldsymbol{D}_S$, gdje $\boldsymbol{D}_A, \boldsymbol{D}_B, \boldsymbol{D}_s$ predstavljaju slučajne varijable koje odgovaraju čvorovima iz skupova A, B i S. Neka je $A', B' \subseteq V$, tako da su A', B' i S disjunktni i $A \subseteq A'$, $B \subseteq B'$, $A' \cup B' \cup S = V$, $A' \perp B'|S$. Promatramo koje se varijable pojavljuju uz $\boldsymbol{D}_{A'}$ kao argumenti faktora ϕ_C. Pritom sa C označimo čvorove klike u grafu G koji odgovaraju slučajnim varijablama $\boldsymbol{D}_{\phi_C}$. Označimo s $C_1 = \{\phi_C \in \Phi |\, A' \cap C \neq \emptyset\}$. Budući da su klike potpuni grafovi, vrijedi: $i \in C, A' \cap C \neq \emptyset \Rightarrow i \in cl(A')$. Pritom je $cl(A')$ definiran kao $cl(A') = A' \cup bd(A')$, a $bd(A') = \cup_{v \in A'} susjedi(A') \setminus A'$. Na temelju definicije uvjetne nezavisnosti, S razdvaja A' od B' u grafu G, te vrijedi $bd(A') \subseteq S \Rightarrow C \subseteq A' \cup S, \forall C \in C_1$. Analogno vrijedi $C \subseteq B' \cup S, \forall C \in C \setminus C_1$. Odavde možemo pisati $\prod_{\phi_C \in \Phi} \phi_C\left(\boldsymbol{d}_{\phi_C}\right) = \prod_{\phi_C \in C_1} \phi_C\left(\boldsymbol{d}_{\phi_C}\right) \cdot \prod_{\phi_C \in C \setminus C_1} \phi_C\left(\boldsymbol{d}_{\phi_C}\right) = \psi_1(\boldsymbol{d}_{A'}, \boldsymbol{d}_S)\psi_2(\boldsymbol{d}_{B'}, \boldsymbol{d}_S)$ čime je pokazano da vrijedi $\boldsymbol{D}_{A'} \perp \boldsymbol{D}_{B'}|\boldsymbol{D}_S$. Marginalizacijom $P(\boldsymbol{y}|\boldsymbol{x})$ preko $\boldsymbol{D}_{A' \setminus A}$ i $\boldsymbol{D}_{B' \setminus B}$ pokazuje se da vrijedi $\boldsymbol{D}_A \perp \boldsymbol{D}_B|\boldsymbol{D}_S$. Na temelju prethodnog izvoda slijedi lokalno Markovljevo svojstvo. Ako se za skup A odabere samo jedan čvor $A = \{v\}$, onda vrijedi $\{v\} \perp V \setminus cl(\{v\})|bd(\{v\})$. Neka su v i v' čvorovi za koje ne vrijedi svojstvo susjedstva $v \sim v'$. Primjenom pravila slabe unije $M \perp (N, P)|Z \Rightarrow M \perp N|(P, Z)$ uz $M = D_v$, $N = D_{v'}, Z = \boldsymbol{D}_{susjedi(v)}$, $P = \boldsymbol{D}_{V \setminus (\{v'\} \cup cl(\{v\}))}$, dobiva se $D_v \perp D_{v'}|D_{V \setminus \{v, v'\}}$. Pritom je iskorištena činjenica da $V \setminus (\{v'\} \cup cl(\{v\})) \cup susjedi(v) = V \setminus \{v, v'\}$. Time je pokazano kako su dva nesusjedna čvora uvjetno nezavisna uz zadane sve ostale čvorove grafa.

Bitno je istaknuti razliku između susjedstva u grafu i susjedstva piksela. Naime pikseli u slici imaju četiri ili osam susjeda ovisno o definiciji susjedstva, dok čvor koji modelira semantičku oznaku i-tog piksela može imati više susjeda. Susjedi čvoru Y_i

mogu primjerice biti svi ostali čvorovi Y_j u grafu G. U tom slučaju čvorovi $\boldsymbol{Y}$ čine potpuno povezani graf.

Budući da uvjetno slučajno polje modelira uvjetnu distribuciju $P(\boldsymbol{Y}|\boldsymbol{X})$, može se promatrati kao djelomično usmjereni graf, u kojem postoji neusmjereni dio definiran na $\boldsymbol{Y}$, koji za roditelje ima varijable iz $\boldsymbol{X}$. Slika 4 prikazuje primjer uvjetnog slučajnog polja.



Slika 4 Primjer uvjetnog slučajnog polja u kojem čvor Y_i koji odgovara i-tom pikselu ima četiri susjeda. U drugačije definiranom uvjetnom slučajnom polju čvor Y_i može imati i puno više susjeda.

2.4. Diskriminativni i generativni modeli

Generativni i diskriminativni modeli opisuju distribucije nad slučajnim varijablama $(\boldsymbol{X}, \boldsymbol{Y})$, ali djeluju u suprotnim smjerovima. Generativni model je familija zajedničkih distribucija koja se faktorizira kao $P(\boldsymbol{Y}, \boldsymbol{X}) = P(\boldsymbol{Y})P(\boldsymbol{X}|\boldsymbol{Y})$, tj. opisuje kako generirati vrijednosti značajki $\mathbf{x}$ uz zadane oznake $\boldsymbol{y}$. Takvi modeli izravno modeliraju zajedničku vjerojatnost $P(\boldsymbol{Y}, \boldsymbol{X})$. Za razliku od njih, diskriminativni modeli izravno modeliraju uvjetnu vjerojatnost $P(\boldsymbol{Y}|\boldsymbol{X})$, dakle samo klasifikacijsko pravilo na temelju kojeg se dodjeljuju semantički razredi pikselima. Prednost ovakvih modela u odnosu na generativne je manji broj parametara budući da se ne modelira $P(\boldsymbol{X})$ koji nije potreban za klasifikaciju, tj. semantičku segmentaciju.

Uvjetna slučajna polja spadaju u kategoriju diskriminativnih modela, te su kao takva vrlo pogodna za semantičku segmentaciju. Za razliku od njih, Markovljeva slučajna polja predstavljaju generativni model koji izravno modelira zajedničku vjerojatnost. Zbog toga se Markovljeva slučajna polja između ostalog primjenjuju i za generiranje

tekstura, te rekonstrukciju slika budući da modeliraju apriornu distribuciju, koja nije nužna za semantičku segmentaciju.

2.5. Zaključivanje i učenje

Dva ključna koraka kod primjene uvjetnih slučajnih polja su učenje i zaključivanje. Proces učenja svodi se na estimaciju parametara uvjetnog slučajnog polja. Konkretni parametri ovise o definiciji uvjetnog slučajnog polja, odnosno njegovih faktora. Ukoliko se koristi definicija zasnovana na potencijalima ψ, parametri uvjetnog slučajnog polja odgovaraju parametrima navedenih potencijala.

Učenje parametara $\boldsymbol{\theta}$ tipično se izvodi primjenom procjene najveće izglednosti $\boldsymbol{\theta} = \underset{\boldsymbol{\theta'}}{\operatorname{argmax}} P(\boldsymbol{Y}_D|\boldsymbol{X}_D, \boldsymbol{\theta'})$, gdje $\{\boldsymbol{X}_D, \boldsymbol{Y}_D\}$ predstavljaju primjere za učenje. U kontekstu semantičke segmentacije, za svaku sliku iz skupa za učenje poznata je njezina semantička segmentacija, te se parametri odabiru tako da model osigurava najveću vjerojatnost za viđene primjere iz skupa za učenje.

Pretpostavimo da se koristi sljedeća definicija uvjetnog slučajnog polja, gdje smo sa $\boldsymbol{\theta}$ označili parametre modela:

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \frac{1}{Z(\boldsymbol{X}|\boldsymbol{\theta})} e^{-\sum_{\psi \in \Psi} \psi(\boldsymbol{X}_\psi, \boldsymbol{Y}_\psi|\boldsymbol{\theta})} \tag{23}$$

U tom slučaju logaritam izglednosti poprimio bi sljedeći oblik:

$$\begin{aligned} \ln L &= \ln \left[\prod_{i=1}^{|D|} \frac{1}{Z(\boldsymbol{X^i}|\boldsymbol{\theta})} e^{-\sum_{\psi \in \Psi} \psi\left(\boldsymbol{X}_\psi^{\boldsymbol{i}}, \boldsymbol{Y}_\psi^{\boldsymbol{i}}|\boldsymbol{\theta}\right)} \right] \\ &= -\sum_{i=1}^{|D|} \ln Z\left(\boldsymbol{X}^i|\boldsymbol{\theta}\right) - \sum_{i=1}^{|D|} \sum_{\psi \in \Psi} \psi\left(\boldsymbol{X}_\psi^{\boldsymbol{i}}, \boldsymbol{Y}_\psi^{\boldsymbol{i}}|\boldsymbol{\theta}\right) \end{aligned} \tag{24}$$

Pritom D predstavlja skup primjera za učenje, dok $|D|$ označava ukupan broj primjera za učenje. Deriviranjem izraza (24) po parametru θ_k dobije se sljedeći izraz:

$$\frac{\partial \ln L}{\partial \theta_k} = -\sum_{i=1}^{|D|} \frac{1}{Z\left(\boldsymbol{X}^i|\boldsymbol{\theta}\right)} \frac{\partial Z\left(\boldsymbol{X}^i|\boldsymbol{\theta}\right)}{\partial \theta_k} - \sum_{i=1}^{|D|} \sum_{\psi \in \Psi} \frac{\partial \psi\left(\boldsymbol{X}_\psi^{\boldsymbol{i}}, \boldsymbol{Y}_\psi^{\boldsymbol{i}}|\boldsymbol{\theta}\right)}{\partial \theta_k} \tag{25}$$

Najveći problem kod primjene izraza (25) jest u izračunu derivacije normalizacijske konstante Z koja uključuje zbroj po svim mogućim $\boldsymbol{y}$. Navedeni problem ovisi o strukturi uvjetnog slučajnog polja, te se za jednostavnija polja može efikasno riješiti.

Međutim, problem se javlja kod polja sa vrlo velikim susjedstvima (veliki skup faktora sa složenim potencijalima) kada se ne mogu efikasno primijeniti klasični algoritmi temeljeni na grafovima. Izračun normalizacijske konstante također se ne može jednostavno provesti ni u slučaju velikog skupa mogućih vrijednosti varijabli Y_i.

Drugi korak kod primjene uvjetnih slučajnih polja jest zaključivanje. U kontekstu semantičke segmentacije, cilj zaključivanja jest da se nakon učenja, na dosad neviđenoj slici svaki piksel klasificira u jedan semantički razred. Zaključivanje se tipično izvodi primjenom MAP (*engl. maximum a posterior probability*) procjenitelja. Cilj je pronaći onu mapu semantičkih oznaka koja maksimizira aposteriornu vjerojatnost (26).

$$\boldsymbol{y}^* = \underset{\boldsymbol{y}}{\operatorname{argmax}}\, P(\boldsymbol{y}|\boldsymbol{x}) \tag{26}$$

Analogno kao i kod učenja, proces zaključivanja ovisi o strukturi uvjetnog slučajnog polja. Iako postoje algoritmi koji omogućavaju egzaktno zaključivanje, računski su dosta zahtjevni, te nisu primjenjivi na složenije modele (modele sa velikim klikama, odnosno velikim brojem klika; primjerice potpuno povezane grafičke modele). Zbog toga su razvijene aproksimativne metode zaključivanja, koje omogućavaju efikasnu izvedbu zaključivanja čak i kod složenijih modela. Jedna takva metoda, temeljena na srednjem polju opisana je u narednim poglavljima.

3. Metoda srednjeg polja

Metoda srednjeg polja spada u kategoriju varijacijskih metoda. Koristi se za aproksimativno zaključivanje kod uvjetnih slučajnih polja. Varijacijske metode su metode optimizacije energijskih funkcionala. Generalna ideja takvih metoda svodi se na uvođenje varijacijskih parametara koji povećavaju broj stupnjeva slobode preko kojih optimiziramo. Spomenuti varijacijski parametri definiraju aproksimacije funkcionala koji se žele optimizirati. Svaki izbor navedenih parametara predstavlja aproksimativno rješenje optimizacijskog problema. Cilj je optimizirati varijacijske parametre kako bi se dobila što bolja aproksimacija [1]. Glavna ideja varijacijskih metoda svodi se na izbor familije distribucija opisane vlastitim parametrima, te pronalazak onih parametara koji odabranu distribuciju čine što sličnijom izvornoj distribuciji. U okviru metode srednjeg polja, varijacijski parametri opisuju distribuciju Q, koju želimo odrediti tako da što bolje aproksimira izvornu distribuciju P.

Promatrajmo općenitu faktoriziranu distribuciju:

$$P(\boldsymbol{X}) = \frac{1}{Z}\prod_{\phi\in\Phi}\phi(\boldsymbol{X}_\phi) \tag{27}$$

Pritom Φ označava skup faktora ϕ koji opisuju distribuciju P. $\boldsymbol{X}$ je vektor diskretnih slučajnih varijabli s malim skupom mogućih vrijednosti, a Z particijska funkcija, koja osigurava da izraz (27) predstavlja vjerojatnosnu distribuciju. $\boldsymbol{X}_\phi$ je skup slučajnih varijabli nad kojima je definiran faktor ϕ. Faktori ϕ mogu predstavljati uvjetne vjerojatnosne distribucije kod Bayesove mreže (u tom slučaju normalizacijska konstanta Z nije potrebna), ili pak potencijale Markovljeve mreže.

Osnovna ideja metode srednjeg polja jest izbjeći zaključivanje koje izravno koristi distribuciju $P(\boldsymbol{X})$ budući da je računanje marginalnih distribucija iz $P(\boldsymbol{X})$ računski zahtjevno. Problem je ogromna dimenzionalnost prostora mogućih realizacija vektora $\boldsymbol{X}$. Promotrimo to na jednostavnom primjeru općenite Bayesove mreže. Neka je Bayesova mreža definirana nad skupom varijabli $\{X_1, X_2, \dots, X_n\}$, te nas zanima $P(x_1|X_2 = 1)$. Izračun spomenute uvjetne vjerojatnosti iziskuje zbroj po svim mogućim kombinacijama vrijednosti $n-2$ preostale slučajne varijable, dakle $P(x_1|X_2 = 1) = \sum_{x_3,x_4,\dots x_n} P(x_1, 1, x_3, x_4, \dots, x_n)$. U slučaju binarnih varijabli, složenost naivnog izračuna postaje eksponencijalna ($O(2^n)$).

3.1. Energijski funkcional

Metoda srednjeg polja nastoji pronaći distribuciju Q, koja omogućava efikasno zaključivanje, te istovremeno veoma dobro aproksimira izvornu distribuciju P. Pritom udaljenost između distribucija P i Q modelira pomoću relativne entropije (Kullback-Leiblerove divergencije). Relativna entropija opisuje količinu informacije koja se izgubi kada se distribucija P aproksimira pomoću distribucije Q.

$$D(Q||P) = \sum_{\boldsymbol{x}} Q(\boldsymbol{x}) \ln \frac{Q(\boldsymbol{x})}{P(\boldsymbol{x})} \tag{28}$$

Zbroj u izrazu (28) obavlja se po svim mogućim vrijednostima $\boldsymbol{x}$ slučajnog vektora $\boldsymbol{X}$. Ako $\boldsymbol{X}$ sadrži n slučajnih varijabli, koje mogu poprimiti neku od v vrijednosti, onda je broj mogućih realizacija vektora $\boldsymbol{X}$ jednak v^n. Zbroj po svim realizacijama $\boldsymbol{x}$ u praksi nije moguće izračunati. U kontekstu teorije informacije relativna entropija (28) tumači se kao očekivani broj dodatnih bitova potrebnih za kodiranje uzoraka od P, kodom prilagođenim za Q umjesto P [1].

Relativna entropija je uvijek nenegativna. U okviru metode srednjeg polja koristi se za ocjenu udaljenosti dviju distribucija, međutim u matematičkom smislu, relativna entropija nije udaljenost budući da ne zadovoljava uvjet simetričnosti $D(P||Q) \neq D(Q||P)$.

Relativna entropija $D(Q||P)$, može se izraziti preko energijskog funkcionala $F[\tilde{P}, Q]$ uzimajući u obzir faktorizaciju (27) distribucije P na sljedeći način:

$$D(Q||P) = \ln Z - F[\tilde{P}, Q] \tag{29}$$

Pritom $\tilde{P}$ predstavlja nenormaliziranu distribuciju P:

$$\tilde{P} = \prod_{\phi \in \Phi} \phi(\boldsymbol{X}_\phi) \tag{30}$$

Energijski funkcional $F[\tilde{P}, Q]$ definiran je kao:

$$F[\tilde{P}, Q] = E_Q\left[\sum_{\phi \in \Phi} \ln \phi(\boldsymbol{X}_\phi)\right] + H_Q(\boldsymbol{X}) = \sum_{\phi \in \Phi} E_Q\left[\ln \phi(\boldsymbol{X}_\phi)\right] + H_Q(\boldsymbol{X}) \tag{31}$$

Izvod izraza (29) detaljnije je prikazan u nastavku. Relativna entropija $D(Q||P)$ može se jednostavnije prikazati pomoću očekivanja:

$$D(Q||P) = \sum_{\boldsymbol{x}} Q(\boldsymbol{x}) \ln \frac{Q(\boldsymbol{x})}{P(\boldsymbol{x})} = \sum_{\boldsymbol{x}} Q(\boldsymbol{x}) \ln Q(\boldsymbol{x}) - \sum_{\boldsymbol{x}} Q(\boldsymbol{x}) \ln P(\boldsymbol{x}) \\ = E_Q[\ln Q(\boldsymbol{X})] - E_Q[\ln P(\boldsymbol{X})] \tag{32}$$

Uzimajući u obzir faktorizaciju distribucije P i svojstvo linearnosti očekivanja, izraz (32) se može detaljnije raspisati kao:

$$\begin{aligned} D(Q||P) &= E_Q[\ln Q(\boldsymbol{X})] - E_Q[\ln P(\boldsymbol{X})] \\ &= E_Q[\ln Q(\boldsymbol{X})] - E_Q\left[\ln \frac{1}{Z} \prod_{\phi \in \Phi} \phi(\boldsymbol{X}_\phi)\right] \\ &= E_Q[\ln Q(\boldsymbol{X})] - E_Q\left[-\ln Z + \sum_{\phi \in \Phi} \ln \phi(\boldsymbol{X}_\phi)\right] \\ &= E_Q[\ln Q(\boldsymbol{X})] + \ln Z - E_Q\left[\sum_{\phi \in \Phi} \ln \phi(\boldsymbol{X}_\phi)\right] \\ &= -H_Q(\boldsymbol{X}) - E_Q\left[\sum_{\phi \in \Phi} \ln \phi(\boldsymbol{X}_\phi)\right] + \ln Z = -F[\tilde{P}, Q] + \ln Z \end{aligned} \tag{33}$$

Pritom je sa $H_Q(\boldsymbol{X})$ definirana entropija od Q:

$$H_Q(\boldsymbol{X}) = -E_Q[\ln Q(\boldsymbol{X})] = -\sum_{\boldsymbol{x}} Q(\boldsymbol{x}) \ln Q(\boldsymbol{x}) \tag{34}$$

Bitno je uočiti da $\ln Z$ ne ovisi o Q zbog čega je minimizacija relativne entropije $D(Q||P)$ ekvivalentna maksimizaciji energijskog funkcionala $F[\tilde{P}, Q]$.

Energijski funkcional sadrži dva člana. Prvi, energijski član $W[\tilde{P}, Q] = \sum_{\phi \in \Phi} E_Q[\ln \phi(\boldsymbol{X}_\phi)]$, uključuje očekivanja logaritama faktora u Φ. Ako su faktori maleni, onda se operator očekivanja primjenjuje samo na maleni skup varijabli. Drugi član predstavlja entropiju distribucije Q i naziva se entropijski član.

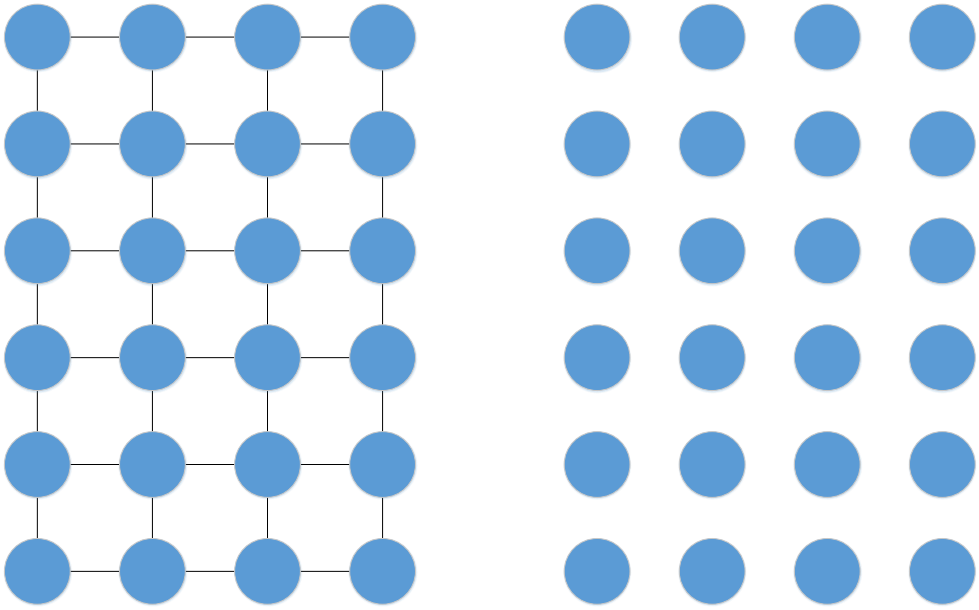
Postavlja se pitanje zašto kod definicije energijskog funkcionala nije korištena relativna entropija $D(P||Q)$. Primjena ovog oblika relativne entropije rezultirala bi pojavom očekivanja po distribuciji P u izrazu (32) što se želi izbjeći budući da je navedena distribucija složena te se u tom slučaju maksimizacija energijskog funkcionala ne bi mogla jednostavno provesti.

3.2. Naivna metoda srednjeg polja

Naivna metoda srednjeg polja nastoji distribuciju P aproksimirati distribucijom Q koja se može prikazati kao produkt nezavisnih marginalnih distribucija .

$$Q(\boldsymbol{X}) = \prod_i Q_i(X_i) \tag{35}$$

Očigledno je da se prilikom ovakve aproksimacije gubi određena razina informacije, ali jednostavnija distribucija Q bitno olakšava zaključivanje kod uvjetnih slučajnih polja budući da se zaključivanje efektivno svede na određivanje distribucije Q. Slika 5 prikazuje primjer izvorne distribucije (lijevi graf) i njezine aproksimacije pomoću naivne pretpostavke (desni graf).



Slika 5 Naivna pretpostavka srednjeg polja. Lijevi graf prikazuje izvornu distribuciju P dok desni graf prikazuje aproksimativnu distribuciju Q.

U nastavku se izvodi iterativni algoritam aproksimativne metode srednjeg polja.

Uzimajući u obzir distribuciju Q (35), energijski član energijskog funkcionala može se zapisati u jednostavnijem obliku (36). Pritom $\boldsymbol{x}_\phi$ predstavlja realizaciju vektora slučajnih varijabli $\boldsymbol{X}_\phi$, tj. $Q(\boldsymbol{x}_\phi) = Q(\boldsymbol{X}_\phi = \boldsymbol{x}_\phi)$.

$$\begin{aligned} W[\tilde{P}, Q] &= \sum_{\phi \in \Phi} E_Q[\ln \phi(\boldsymbol{X}_\phi)] = \sum_{\phi \in \Phi} \sum_{\boldsymbol{x}_\phi} Q(\boldsymbol{x}_\phi) \ln \phi(\boldsymbol{x}_\phi) \\ &= \sum_{\phi \in \Phi} \sum_{\boldsymbol{x}_\phi} \left(\prod_{X_i \in \boldsymbol{X}_\phi} Q_i(x_i) \right) \ln \phi(\boldsymbol{x}_\phi) \end{aligned} \tag{36}$$

Analognim razmatranjem može se entropija zapisati kao:

$$H_Q(\boldsymbol{X}) = -E_Q[\ln Q(\boldsymbol{X})] = -E_Q\left[\ln \prod_i Q_i(X_i)\right] = -E_Q\left[\sum_i \ln Q_i(X_i)\right] \\ = -\sum_i E_Q[\ln Q_i(X_i)] = \sum_i H_Q(X_i) \tag{37}$$

Cilj metode srednjeg polja jest pronaći $\{Q_i(X_i)\}$ koji minimizira $D(Q||P)$ odnosno maksimizira $F[\tilde{P}, Q]$ uz ograničenje $\forall i \ \sum_{x_i} Q(x_i) = 1$. Problem se formalno može definirati na sljedeći način:

$$Q = \underset{\{Q_i(X_i)\}}{\operatorname{argmax}} F[\tilde{P}, Q] \ \ tako\ da: \\ Q(\boldsymbol{X}) = \prod_i Q_i(X_i) \\ \sum_{x_i} Q_i(x_i) = 1, \forall i \tag{38}$$

Za potrebe optimizacije definira se Lagrangeova funkcija L:

$$L[Q] = \sum_{\phi \in \Phi} E_Q\left[ln\phi(\boldsymbol{X}_\phi)\right] + \sum_j H_Q(X_j) + \sum_j \lambda_j \left(\sum_{x_j} Q_j(x_j) - 1\right) \tag{39}$$

Multiplikator λ_j odgovara uvjetu da je $Q_j(X_j)$ vjerojatnosna distribucija. S ciljem određivanja maksimuma energijskog funkcionala potrebno je derivirati izraz (39) po svim $Q_i(x_i)$. U prethodnoj notaciji, prvi indeks i označava distribuciju Q_i koja odgovara i-tom pikselu (takvih distribucija ima onoliko koliko i piksela), dok drugi indeks i ne ovisi o pikselu, nego se koristi samo za označavanje realizacije varijable X_i. Drugim riječima, x_i predstavlja jednu moguću vrijednost varijable X_i. Derivacije je potrebno provesti po svim mogućim vrijednostima od x_i. U narednim izrazima x_i je fiksiran na jednu moguću vrijednost od X_i. Izraz (40) prikazuje derivaciju energijskog člana, a izraz (41) derivaciju entropijskog člana energijskog funkcionala. Konačnu derivaciju Lagrangeove funkcije prikazuje izraz (42) [1].

$$\frac{\partial}{\partial Q_i(x_i)}\left[\sum_{\phi\in\Phi}\sum_{\boldsymbol{x}_\phi}\left(\prod_{X_j\in\boldsymbol{X}_\phi} Q_j(x_j)\right)\ln\phi(\boldsymbol{x}_\phi)\right]$$

$$= \sum_{\phi\in\Phi}\sum_{\boldsymbol{x}_\phi}\frac{\partial}{\partial Q_i(x_i)}\prod_{X_j\in\boldsymbol{X}_\phi} Q_j(x_j)\ln\phi(\boldsymbol{x}_\phi) \tag{40}$$

$$= \sum_{\phi\in\Phi}\sum_{\boldsymbol{x}_\phi} Q(\boldsymbol{x}_\phi|x_i)\ln\phi(x_\phi) = \sum_{\phi\in\Phi} E_Q\left[\ln\phi(\boldsymbol{X}_\phi)\,|x_i\right]$$

$$\frac{\partial}{\partial Q_i(x_i)}\left[\sum_j H_Q(X_j)\right] = \frac{\partial}{\partial Q_i(x_i)}\left[-\sum_j\sum_{x_j} Q_j(x_j)\ln Q_j(x_j)\right] \tag{41}$$

$$= -\ln Q_i(x_i) - 1$$

$$\frac{\partial}{\partial Q_i(x_i)}L[Q] = \sum_{\phi\in\Phi} E_Q\left[\ln\phi(\boldsymbol{X}_\phi)\,|x_i\right] - \ln Q_i(x_i) - 1 + \lambda_i \tag{42}$$

Izjednačavanjem izraza (42) s nulom dobije se:

$$\ln Q_i(x_i) = \lambda_i - 1 + \sum_{\phi\in\Phi} E_Q\left[\ln\phi(\boldsymbol{X}_\phi)\,|x_i\right] \tag{43}$$

Eksponenciranjem jednadžbe (43) i renormalizacijom dobije se $Q_i(X_i)$ (44) koji predstavlja optimalno rješenje problema (38) za zadani $\{Q_j(X_j)\}_{j\neq i}$. Prilikom renormalizacije λ_i se izgubi budući da je konstanta s obzirom na x_i.

$$Q_i(x_i) = \frac{1}{Z_i}e^{\sum_{\phi\in\Phi} E_Q[\ln\phi(\boldsymbol{X}_\phi)|x_i]}$$
$$Z_i = \sum_{x_i} e^{\sum_{\phi\in\Phi} E_Q[\ln\phi(\boldsymbol{X}_\phi)|x_i]} \tag{44}$$

Jednadžba (44) predstavlja maksimum funkcionala $F[\tilde{P}, Q]$ uz zadane sve $Q_j(X_j), j \neq i$. Spomenuti funkcional definiran je kao zbroj dva člana: $\sum_{\phi\in\Phi} E_Q[ln\phi(\boldsymbol{X}_\phi)]$, koji je linearan s obzirom na $Q_i(X_i)$ uz zadane sve $Q_j(X_j), j \neq i$, i $\sum_j H_Q(X_j)$, koji je konkavna funkcija s obzirom na $Q_i(X_i)$ uz zadane sve $Q_j(X_j), j \neq i$. Kao cjelina $F[\tilde{P}, Q]$ je konkavna funkcija s obzirom na $Q_i(X_i)$ za zadane sve $Q_j(X_j), j \neq i$, te kao takva ima jedan globalni maksimum (44) [1].

3.3. Iterativni algoritam srednjeg polja

U ovom poglavlju izlaže se općeniti algoritam srednjeg polja. Algoritam je iterativnog i aproksimativnog karaktera. Jednadžba (44) predstavlja temeljnu jednadžbu

osvježavanja navedenog algoritma. Ako slučajna varijabla X_i nije element vektora $\boldsymbol{X}_\phi$ onda $E_Q[\ln \phi(\boldsymbol{X}_\phi)\,|x_i]$ prelazi u $E_Q[\ln \phi(\boldsymbol{X}_\phi)]$. Dobiveno očekivanje nije ovisno o X_i, te se može ubaciti u normalizacijsku konstantu Z_i pri čemu se dobiva sljedeće pojednostavljenje izraza (44):

$$Q_i(x_i) = \frac{1}{Z_i} e^{\sum_{\phi: X_i \in \boldsymbol{X}_\phi} E_{(\boldsymbol{X}_\phi - \{X_i\}) \sim Q}[\ln \phi(\boldsymbol{X}_\phi, x_i)]} \tag{45}$$

U prethodnom izrazu (45) $E_{(\boldsymbol{X}_\phi - \{X_i\}) \sim Q}$ označava očekivanje slučajnih varijabli $\boldsymbol{X}_\phi$ ne uključujući slučajnu varijablu X_i, dok $\phi(\boldsymbol{X}_\phi, x_i)$ predstavlja vrijednost faktora ϕ kada slučajna varijabla X_i poprimi vrijednost x_i. Tablica 1 prikazuje općeniti aproksimativni algoritam srednjeg polja temeljen na izrazu (45).

Tablica 1 Općeniti algoritam srednjeg polja

1. *Inicijalizacija:* $Q \leftarrow Q_0$
2. *Ponavljaj do konvergencije:*
   a. *Za svaki* Q_i *ponovi:*
      i. Za svaki $x_i \in Val(X_i)$:
         1. $Q_i(x_i) \leftarrow \exp\left\{\sum_{\phi: X_i \in \boldsymbol{X}_\phi} E_{(\boldsymbol{X}_\phi - \{X_i\}) \sim Q}[\ln \phi(\boldsymbol{X}_\phi, x_i)]\right\}$
      ii. Normaliziraj $Q_i(X_i)$ tako da $\sum_{x_i \in Val(X_i)} Q_i(x_i) = 1$

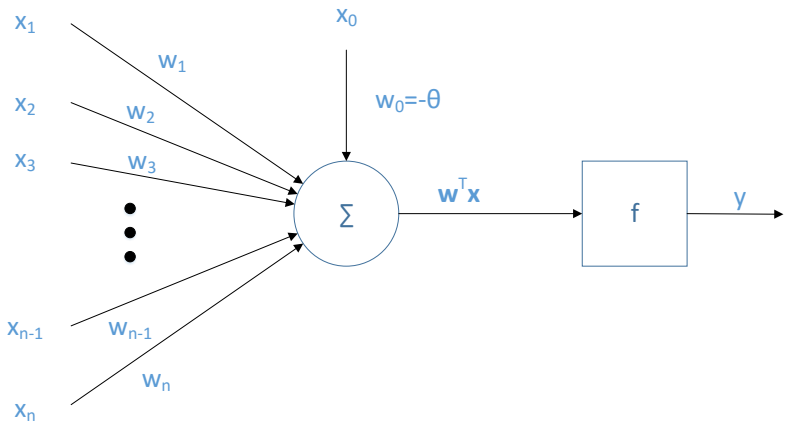
$Val(X_i)$ označava skup svih vrijednosti koje slučajna varijabla X_i može poprimiti. U praktičnim implementacijama uvjet konvergencije se najčešće definira kao zadani broj iteracija. Uvjet je također moguće formulirati kao maksimalnu dopuštenu razliku distribucija Q između dva uzastopna koraka algoritma.

4. Umjetne neuronske mreže

Umjetne neuronske mreže su porodica modela u umjetnoj inteligenciji i strojnom učenju koji nastoje simulirati način učenja i zaključivanja bioloških organizama. Ljudski neuron i način rada ljudskog mozga inspirirao je nastanak umjetnih neurona i neuronskih mreža koje se danas uspješno primjenjuju za rješavanje mnoštva problema presloženih za klasični algoritamski pristup. Jedan takav problem jest problem semantičke segmentacije za koji su najbolji rezultati dobiveni primjenom konvolucijskih neuronskih mreža [6][9][16]. U ovom poglavlju dan je kratki pregled umjetnih neurona, neuronskih mreža, te načina njihovog učenja.

4.1. Umjetni neuron

Umjetni neuron je osnovna građevna komponenta neuronske mreže. Prvi model neurona bio je McCulloch-Pittsov umjetni neuron [17], koji simulira način rada pojednostavljenog biološkog neurona. Informacija u neuronu predstavljena je kao razlika potencijala između unutrašnjeg i vanjskog dijela stanice [18]. Svaki neuron prima signale (električne impulse) od susjednih neurona, mijenjajući vlastiti potencijal. Kada ukupni napon prijeđe određeni prag, neuron se aktivira te generira električni impuls koji šalje susjednim neuronima. Opisano pojednostavljeno ponašanje biološkog neurona modelira sljedeći model (slika 6).



Slika 6 Model umjetnog neurona

Izlaz neurona definiran je kao:

$$y = f\left(\sum_{i=0}^{n} w_i x_i\right) = f(\boldsymbol{w}^T \boldsymbol{x}) \tag{46}$$

Pritom su w_i težine koje modeliraju jakost sinapsi (spojeva između bioloških neurona), a x_i signali primljeni od susjednih neurona. θ predstavlja prethodno opisani prag neurona, te se radi kraćeg zapisa pomoću skalarnog produkta definira da je $x_0 = 1$ te $w_0 = -\theta$. f je aktivacijska funkcija, koja je u slučaju McCulloch-Pittsovog neurona definirana kao funkcija praga:

$$f(\boldsymbol{w}^T\boldsymbol{x}) = \begin{cases} 1, & \boldsymbol{w}^T\boldsymbol{x} \geq 0 \\ 0, & inače \end{cases} \tag{47}$$

Učenje neurona svodi se na određivanje vrijednosti njegovih težina $\boldsymbol{w}$. U slučaju jednog neurona bez aktivacijske funkcije, za učenje se može koristiti LMS algoritam (*engl. least mean squares*). Učenje se provodi u nadziranoj konfiguraciji, gdje je za svaki ulaz x poznat željeni izlaz t. Cilj učenja je smanjiti srednju kvadratnu pogrešku $E = \frac{1}{2N}\sum_{i=1}^{N}\left(t^{(i)} - y^{(i)}\right)^2$. Pritom N predstavlja broj primjera za učenje, a $y^{(i)}$ odziv neurona na i-ti ulaz. Pseudokod algoritma prikazuje tablica 2.

Tablica 2 Algoritam učenja perceptrona

Postavke:

- i – indeks iteracije
- $x^{(i)}, y^{(i)}, t^{(i)}$ – ulaz, izlaz, željeni izlaz u i-toj iteraciji
- $w_j^{(i)}$ – težina j-te sinapse u i-toj iteraciji

1. Inicijaliziraj težine $\boldsymbol{w}^{(0)}$
2. $i \leftarrow 0$
3. Ponavljaj do konvergencije:
   a. $i \leftarrow i + 1$
   b. $w_j^{(i)} \leftarrow w_j^{(i-1)} + \eta\left(t^{(i)} - y^{(i)}\right)x_j^{(i)}$

Za minimizaciju srednje kvadratne pogreške koristi se gradijentni spust kako bi se pronašle optimalne vrijednosti težina. Izvod jednadžbe osvježavanja težina prikazan je u nastavku (48). Pritom se umjesto stvarnog gradijenta srednje kvadratne pogreške (gradijent određen na temelju svih primjera u skupu za učenje) koristi gradijent kvadratne pogreške za primjer u trenutnoj iteraciji.

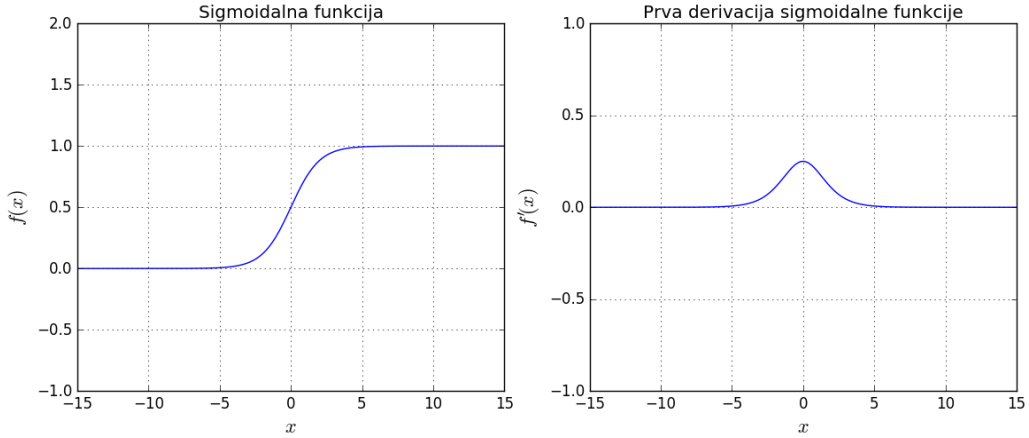
$$w_j = w_j - \eta \frac{dE^{(i)}}{dw_j} = w_j - \eta \frac{d}{dw_j}\left(\frac{1}{2}\left(t^{(i)} - y^{(i)}\right)^2\right)$$
$$= w_j - \eta \frac{1}{2} 2\left(t^{(i)} - y^{(i)}\right)(-1) \frac{d}{dw_j}\left(\sum_{k=0}^{n} w_k x_k\right) \quad (48)$$
$$= w_j + \eta\left(t^{(i)} - y^{(i)}\right) x_j$$

4.1.1. Aktivacijske funkcije

Aktivacijske funkcije su funkcije koje se primjenjuju nad izlazima neurona. Pored prethodno opisane funkcije praga (47) postoji još niz drugih funkcija koje se koriste kao aktivacijske funkcije neurona. Jedna od najčešće korištenih je sigmoidalna funkcija:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (49)$$

Sigmoidalna funkcija je nelinearna i monotono rastuća funkcija (slika 7). Njezina popularnost vezana je uz svojstvo derivabilnosti (50), koje je omogućilo učenje neuronskih mreža pomoću algoritma povratne propagacije pogreške.



Slika 7 Sigmoidalna aktivacijska funkcija(lijevo) i njezina derivacija(desno)

$$\frac{df(x)}{dx} = \frac{d}{dx}\left(\frac{1}{1 + e^{-x}}\right) = \frac{-1}{(1 + e^{-x})^2} e^{-x}(-1) = \frac{e^{-x}}{(1 + e^{-x})^2}$$
$$= \frac{1}{1 + e^{-x}} \frac{1 + e^{-x} - 1}{1 + e^{-x}} = \frac{1}{1 + e^{-x}}\left(1 - \frac{1}{1 + e^{-x}}\right) \quad (50)$$
$$= f(x)[1 - f(x)]$$

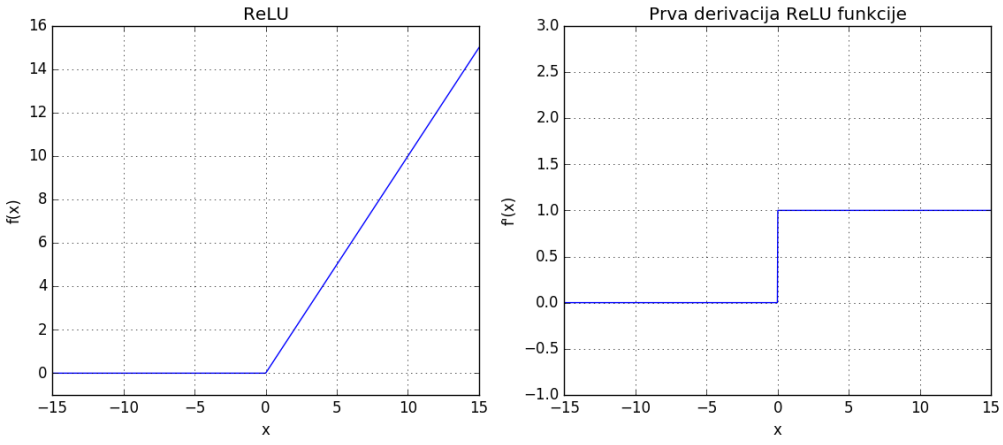
U početnom razdoblju uporabe neuronskih mreža (nakon pojave algoritma povratne propagacije pogreške), sigmoidalne funkcije bile su najčešće korištene aktivacijske funkcije neurona. Međutim, s pojavom dubokih neuronskih mreža, njihov utjecaj je

pao zbog negativnog efekta nestajućih gradijenata (*engl. vanishing gradient*). Sigmoidalna funkcija ograničava izlaz neurona na interval $(0,1)$ pri čemu se asimptotski približava vrijednosti 1 odnosno nula. Problem nestajućih gradijenata predstavlja situaciju kada neuroni u dubljim slojevima mreže postanu pogrešno zasićeni, što znači da aktivacija neurona (vrijednost otežanog zbroja ulaza) postane veoma velika vrijednost pogrešnog predznaka. Posljedica toga jest spora konvergencija algoritma učenja budući da su gradijenti koji utječu na promjenu težina neurona veoma maleni (slika 7).

U dubokim neuronskim mrežama najčešće se kao aktivacijska funkcija koristi ReLU (engl. rectified linear unit). Formalno ReLU funkcija definirana je kao:

$$f(x) = \max(0, x) \tag{51}$$

Glavna prednost ReLU (slika 8) u odnosu na sigmoidalnu funkciju jest manja izglednost pojave nestajućih gradijenata budući da za pozitivni ulaz, gradijent je uvijek različit od nule (i jednak 1). Negativna vrijednost aktivacije neurona za sobom povlači gradijent jednak nuli, koji efektivno gasi promatrani neuron budući da se njegove težine više ne mijenjaju, te se na taj način ostvaruje rijetka reprezentacija mreže.



Slika 8 ReLU aktivacijska funkcija(lijevo) i njezina derivacija(desno)

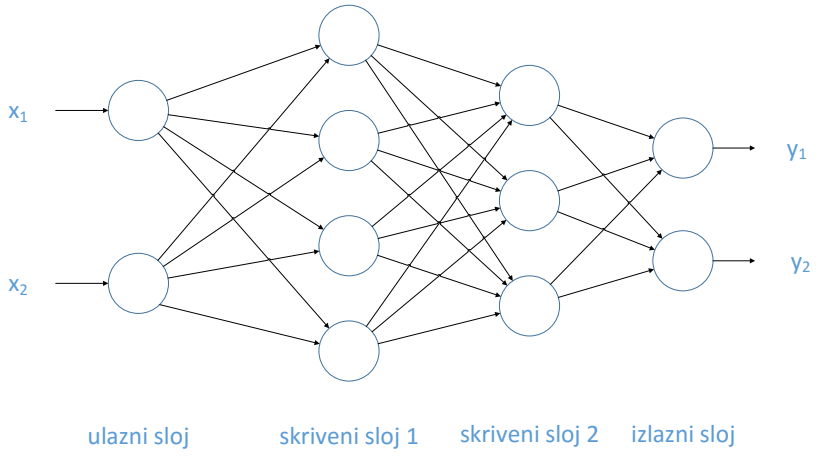
4.2. Neuronske mreže

Umjetni neuron je vrlo jednostavna procesna jedinica, te kao takva nije u stanju sama riješiti kompleksnije probleme. Ako se pogleda osnovni perceptron (neuron s pragom kao aktivacijskom funkcijom), nelinearni problemi već izlaze izvan opsega problema koji se njime mogu riješiti. Iz tog razloga, neuroni se međusobno povezuju

u složenije strukture – neuronske mreže, koje kroz međusobnu interakciju sastavnih dijelova uspješno rješavaju složene probleme. Način na koji su neuroni međusobno povezani određuje arhitekturu neuronske mreže.

4.2.1. Višeslojni perceptron

Višeslojni perceptron je slojevita unaprijedna neuronska mreža (slika 9). Neuroni su organizirani u slojeve, pri čemu mreža sadrži ulazni, izlazni, te jedan ili više skrivenih slojeva. Ulazni neuroni ne obavljaju nikakvu obradu, nego služe samo za predstavljanje ulaznih podataka. Svi ostali neuroni sadrže nelinearne aktivacijske funkcije jer u suprotnom se višeslojni perceptron svodi na običan linearni perceptron zbog svojstva linearnosti aktivacijske funkcije. Višeslojni perceptron predstavlja unaprijednu mrežu budući da ne postoje povratne veze između slojeva.



Slika 9 Višeslojni perceptron

Opisana arhitektura je osnovna vrsta neuronske mreže, koja se često koristi za rješavanje mnogih klasifikacijskih i regresijskih problema. Mnoge druge arhitekture mreža temelje se upravo na višeslojnom perceptronu.

4.2.2. Konvolucijske neuronske mreže

Konvolucijske neuronske mreže su vrsta neuronskih mreža prilagođena obradi slika. Riječ je o slojevitim mrežama s konvolucijskim slojevima. Navedene mreže našle su široku primjenu u raznim područjima računalnog vida, pa tako i u području semantičke segmentacije. Popularnost im je posebno porasla s razvojem grafičkih procesora, koji su omogućili brzo učenje dubokih mreža sa veoma velikim brojem slojeva.

4.2.2.1. Konvolucijski sloj

Konvolucijski sloj predstavlja temeljni građevni blok konvolucijske neuronske mreže. Kao što sam naziv kaže, njegova zadaća je obavljanje konvolucije ulazne slike s naučenom jezgrom filtra. Konvolucijski slojevi, odnosno mreže inspirirane su vidnim korteksom mačaka [20], koji se sastoji od mnoštva ćelija, koje su osjetljive na malene regije u vidnom polju. Navedene regije nazivaju se receptivnim poljima. Svaka spomenuta ćelija u osnovi djeluje kao lokalni filtar, koji detektira karakteristične značajke u ulaznoj slici.

Kod višeslojnog perceptrona tipično su na ulaz svakog neurona spojeni svi neuroni prethodnog sloja. U tom slučaju govori se o potpunoj povezanosti. Ako bi se višeslojni perceptron izravno primjenjivao nad slikama, tako da su slojevi dvodimenzionalni, spomenuta arhitektura rezultirala bi ogromnim brojem parametara. Za sliku dimenzije $100x100$, dakle 10000 ulaznih neurona, 100 skrivenih neurona i 1 izlazni neuron, mreža bi imala $10000 \cdot 100 + 100 + 100 \cdot 1 + 1 = 1000201$ parametara. Učenje navedene mreže zahtijevalo bi značajnu količinu vremena. Prednost konvolucijskih slojeva u odnosu na potpuno povezane slojeve višeslojnog perceptrona jest lokalno receptivno polje i dijeljenje težina.

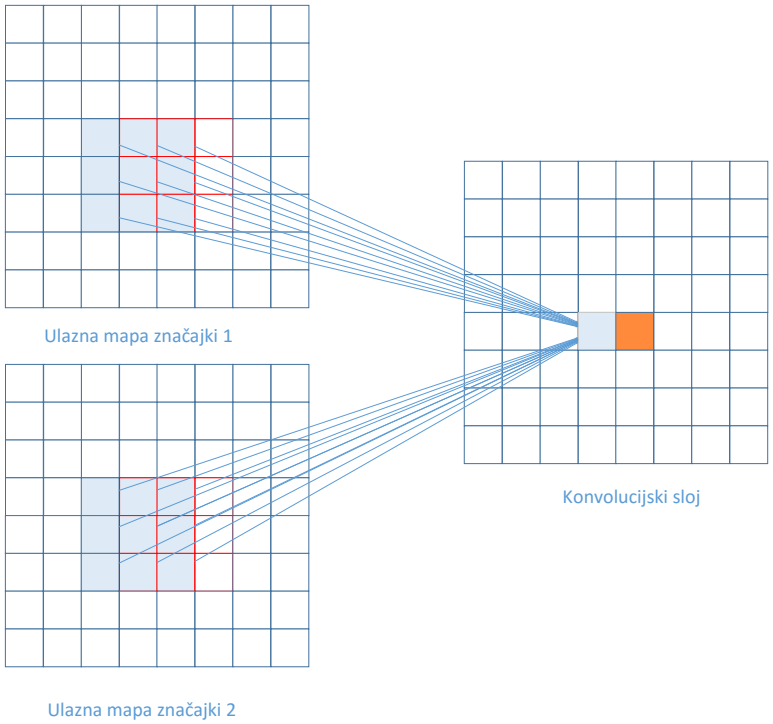
Naime, konvolucijski slojevi definirani su veličinom jezgre filtra, pomakom jezgre, te brojem filtara. U postupku učenja konvolucijska neuronska mreža treba naučiti jezgre filtara, koje najbolje detektiraju značajke u ulaznim slikama. Svaki konvolucijski sloj može definirati više filtara, koji određuju broj izlaznih mapa koje konvolucijski sloj generira. Osnovna ideja konvolucijskih slojeva jest primjena naučenih filtara nad svim ulaznim mapama. Pritom se svaki filtar primjenjuje na svim pozicijama u ulaznoj mapi. Time se ostvaruje detekcija značajke na bilo kojem mjestu u ulaznoj slici. Nasuprot tome, višeslojni perceptron mogao bi detektirati značajku samo na onim mjestima u slici gdje se ona pojavljivala tijekom učenja.

Izlaz svakog neurona unutar konvolucijskog sloja određuje se kao težinski zbroj izlaza onih neurona iz prethodnog sloja, koji se nalaze u receptivnom polju promatranog neurona. Pritom se na dobiveni zbroj može primijeniti odabrana aktivacijska funkcija (52).

$$y_{ij}^{(k)} = f\left(\sum_{o}\sum_{m\in S_{ij}} w_m^{(k)} x_m^{(o)} + w_0^{(k)}\right) \quad (52)$$

Indeks izlazne mape označen je s k. S_{ij} predstavlja skup indeksa neurona prethodnog sloja koji leže u receptivnom polju neurona na poziciji (i, j). $w_m^{(k)}$ su težine pridijeljene k-tom filtru, dok $w_0^{(k)}$ predstavlja pripadni prag. Indeks ulazne mape označen je s o. $x_m^{(o)}$ predstavlja vrijednost m-tog neurona u o-toj ulaznoj mapi. f je odabrana aktivacijska funkcija (najčešće ReLU).

Veličina jezgre određuje veličinu receptivnog polja neurona. Tipične veličine su: $3x3$, $5x5$, $7x7$. U slučaju jezgre $3x3$, svaki neuron spojen je sa 9 susjednih neurona iz svake mape prethodnog sloja (slika 10). Pomak jezgre definira razmak između neurona u konvolucijskom sloju. Na slici je korišten pomak iznosa 1, te su susjedi crvenog neurona obrubljeni crvenom bojom.



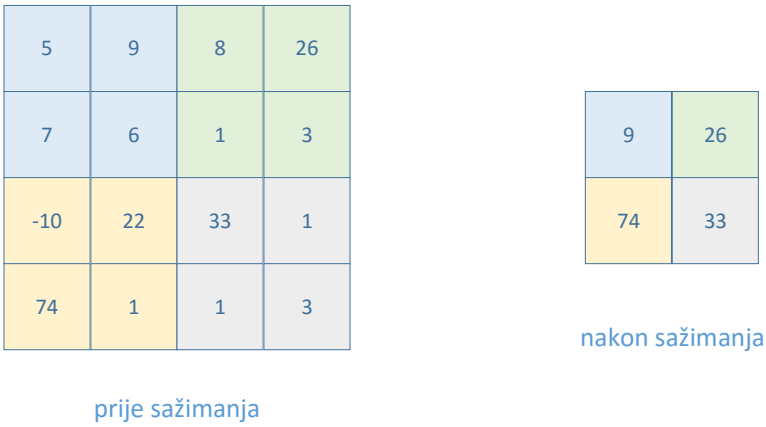
Slika 10 Lokalno receptivno polje neurona

Budući da konvolucijski sloj primjenjuje svaki filtar nad svim pozicijama u ulaznoj mapi, neuroni, koji definiraju isti filtar, imaju jednake vrijednosti težina. Time se drastično smanjuje broj parametara u odnosu na potpuno povezane slojeve, što ubrzava učenje konvolucijskih mreža. Neka je ulaz u konvolucijski sloj dimenzija $100x100$ piksela, te neka sadrži 3 mape (po jedna mapa za svaku boju).

Pretpostavimo da konvolucijski sloj sadrži 4 filtra dimenzija $7x7$. U tom slučaju, učenje navedenog konvolucijskog sloja iziskivalo bi određivanje vrijednosti $7 \cdot 7 \cdot 3 \cdot 4 + 4 = 592$ parametra, što je mnogo manje od prethodno spomenutog broja parametara potpuno povezanog sloja.

4.2.2.2. Sažimajući sloj

Sažimanje je često korištena operacija u konvolucijskim mrežama. Radi se o postupku koji smanjuje dimenzije svojih ulaza obavljajući njihovo pod-uzorkovanje. Postoje različite vrste sažimanja od kojih se najčešće koristi sažimanje odabirom maksimalnog odziva (*engl. max pooling*). Sažimanje započinje podjelom ulaza na podregije zadane veličine. Pritom je preklapanje podregija određeno definiranim pomakom. Slika 11 prikazuje podjelu ulaza na podregije veličine $2x2$ uz pomak 2. Sažimanje odabirom maksimalnog odziva odabire iz svake podregije maksimalnu vrijednost, te nju prosljeđuje na izlaz. Postupak sažimanja odabirom maksimalnog odziva prikazuje slika 11.



Slika 11 Sažimanje odabirom maksimalnog odziva

Posljedica sažimanja je redukcija dimenzionalnosti ulaza, što sa sobom povlači smanjenje broja parametara za učenje. Dodatno, sažimanjem se smanjuje osjetljivost konvolucijske mreže na malene translacije u slici, što je korisno u slučaju malenih promjena položaja značajke koja se nastoji detektirati filtrom konvolucijskog sloja unutar promatranog receptivnog polja.

4.2.2.3. Regularizacija

Prilikom učenja na manjim skupovima podataka, duboke neuronske mreže sklone su pretreniranju zbog velikog kapaciteta kojeg posjeduju. U takvim slučajevima

mreža se prilagođava šumu u skupu za učenje, te gubi dobra svojstva generalizacije na dotad neviđenim podacima. Klasičan pristup rješavanju navedenog problema je regularizacija modela. Osnovna ideja regularizacije jest dodavanje regularizacijskog člana u definiciju funkcije gubitka, koja se želi minimizirati prilikom učenja modela. Spomenuti regularizacijski član, definiran nad parametrima modela (težinama neurona), nastoji potisnuti težine prema nuli, čime se efektivno smanjuje složenost modela, budući da se pojedini neuroni nastoje isključiti. U praksi se najčešće koriste dvije vrste regularizacije: L_1 (53) i L_2 (54). Pritom $\boldsymbol{\theta}$ označava vektor parametara modela, a θ_i i-ti parametar.

$$L_1 = \|\boldsymbol{\theta}\|_1 = \sum_i |\theta_i| \tag{53}$$

$$L_2 = \|\boldsymbol{\theta}\|_2^2 = \sum_i \theta_i^2 \tag{54}$$

Kod dubokih neuronskih mreža popularan je drugačiji pristup regularizaciji koji se temelji na *dropout* metodi [21]. *Dropout* metoda svodi se na nasumično gašenje pojedinih neurona prilikom učenja neuronske mreže. Time se efektivno smanjuje kapacitet mreže prilikom učenja i otežava pretreniranje. *Dropout* se primjenjuje nad pojedinim slojevima neuronske mreže. Pritom se definira vjerojatnost gašenja neurona, koja utječe na prosječan broj neurona koji će biti ugašeni tijekom učenja. Gašenje neurona svodi se na postavljanje vrijednosti njegovog izlaza na nulu u unaprijednom prolazu kroz mrežu, te postavljanje gradijenta na nulu u povratnom prolazu. Vjerojatnost gašenja neurona definira se posredno preko parametra p. Neka je r slučajna varijabla koja se ponaša po Bernoullijevoj distribuciji $\mathcal{B}(p)$. U tom slučaju izlaz neurona prilikom učenja definiran je kao:

$$y = r \cdot f(\boldsymbol{w}^T \boldsymbol{x}) \tag{55}$$

Veći p smanjuje vjerojatnost da će neuron biti ugašen prilikom učenja. Kada se mreža nauči, dropout se u fazi zaključivanja više ne koristi te su svi neuroni aktivni.

Odabir slojeva na koje će se primijeniti *dropout* predstavlja odluku dizajna mreže. Ako se želi izbjeći prenaučenost, poželjno je dropout primijeniti na slojeve s najvećim kapacitetom.

4.2.2.4. Grupna normalizacija

Grupna *(engl. batch)* normalizacija [22] je metoda kojom se nastoji smanjiti promjenjivost distribucija aktivacija neurona u neuronskoj mreži. Naime, prilikom učenja slojevite mreže ulazi u neurone jednog sloja određeni su aktivacijama neurona prethodnog sloja. Budući da se prilikom učenja mijenjaju težine neurona, dolazi do promjene distribucija njihovih aktivacija. Promjena distribucija otežava učenje težina neurona u sljedećim slojevima budući da se težine moraju prilagoditi statističkim svojstvima nove distribucije. Grupna normalizacija rješava opisani problem normalizacijom podataka unutar jedne grupe za učenje (*engl. batch*). Grupna normalizacija vrši se na razini pojedinog neurona.

Neka $x_1, x_2, \dots, x_m$ predstavljaju ulaze neurona u jednoj grupi za učenje veličine m. Normalizacija je transformacija spomenutih ulaza na sljedeći način:

$$\hat{x}_k = \frac{x_k - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{56}$$

Pritom $\hat{x}_k$ ima jediničnu varijancu i srednju vrijednost jednaku nuli. ϵ je maleni broj koji osigurava da je nazivnik različit od nule. Varijanca i srednja vrijednost grupe procjenjuju se na sljedeći način:

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{57}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_B)^2 \tag{58}$$

Međutim, normalizacija opisana izrazom (56) nije dovoljna budući da može promijeniti što neuron može prikazati. Normalizacijom (56) ulaza u sigmoidalnu aktivacijsku funkciju, pripadni neuron se ograničava na linearni dio sigmoidalne funkcije, te ima manju reprezentacijsku moć u odnosu na neuron čija aktivacija nije normalizirana. Kako bi se navedeni problem izbjegao, transformacija (56) proširuje se parametrima β i γ (59), koji se uče tijekom procesa učenja neuronske mreže. Ovisno o situaciji učenje spomenutih parametara može se i izostaviti.

$$\tilde{x}_k = \gamma \hat{x}_k + \beta \tag{59}$$

Grupna normalizacija može se primijeniti u konvolucijskim mrežama tako da se normaliziraju aktivacije konvolucijskih slojeva. Kako bi se osiguralo konvolucijsko svojstvo, normalizacija se radi nad svim elementima mape značajki. Drugim

riječima, prilikom procjene varijance i srednje vrijednosti grupe, uzimaju se u obzir aktivacije na svim lokacijama u mapi značajki, za sve primjere iz grupe. Budući da neuroni, koji definiraju jedan filtar u izlaznoj mapi konvolucijskog sloja dijele težine, navedeni neuroni dijelit će i parametre γ i β. Analogno težinama neurona, parametri γ i β mogu se učiti postupkom povratne propagacije pogreške. Prilikom primjene grupne normalizacije, bitno je uočiti da se varijanca i srednja vrijednost tijekom učenja procjenjuju na temelju ulaznih podataka u grupi za učenje. Procijenjene vrijednosti koriste se za izračun njihovog pomičnog prosjeka. Kada učenje mreže završi, prilikom unaprijednog prolaska dotad neviđenog ulaza kroz mrežu, koriste se prethodno izračunate prosječne vrijednosti varijance i srednje vrijednosti.

Grupna normalizacija često se primjenjuje kod učenja dubokih neuronskih mreža budući da omogućava njihovo brže učenje, te istovremeno djeluje kao regularizator smanjujući potrebu za *dropoutom* [22].

4.2.2.5. Softmax

Softmax funkcija je generalizacija sigmoidalne funkcije koja transformira k-dimenzionalni vektor $\boldsymbol{x}$ s proizvoljnim realnim vrijednostima u k-dimenzionalni vektor $f(\boldsymbol{x})$ čije komponente imaju vrijednosti u intervalu $(0,1)$, te im je zbroj jednak 1:

$$f(\boldsymbol{x})_j = \frac{e^{x_j}}{\sum_{i=1}^{k} e^{x_i}}, j \in \{1,2,\dots,k\}$$
$$\sum_{j=1}^{k} f(\boldsymbol{x})_j = 1 \qquad (60)$$

Opisana funkcija koristi se za probabilističku interpretaciju izlaza neuronskih mreža. Općenito, izlaz neuronske mreže (neuroni zadnjeg sloja imaju funkciju identiteta kao aktivacijsku funkciju) predstavlja stupanj sigurnosti mreže u donesenu odluku. Spomenuti stupanj je realan broj. Veći broj označava veću sigurnost. Kako bi se izlaz neuronske mreže mogao interpretirati kao vjerojatnosna distribucija, na izlaze neurona izlaznog sloja primjenjuje se softmax funkcija. Time se postiže da su izlazi svih izlaznih neurona ograničeni na interval $(0,1)$, te je zbroj svih izlaza jednak 1.

4.2.3. Učenje neuronskih mreža

Nakon pojave umjetnog neurona i algoritma njegovog učenja protekao je značajan niz godina prije nego što su višeslojne neuronske mreže ušle u širu upotrebu.

Naime, postojao je problem s učenjem višeslojnih mreža. Za razliku od perceptrona za koji je poznat njegov željeni izlaz, kod neurona u skrivenom sloju višeslojne mreže, njegov željeni izlaz nije unaprijed poznat što je u početku sprječavalo učenje višeslojnih mreža. Za takve neurone nije bilo moguće odrediti koliko su zaslužni za pogrešku mreže (*engl. credit assignment problem*). Opisani problem riješen je pojavom algoritma povratne propagacije pogreške (*engl. backpropagation algorithm*).

4.2.3.1. Algoritam povratne propagacije pogreške

Kao što je već rečeno, učenje neuronske mreže svodi se na određivanje vrijednosti težina svih neurona u mreži. Pritom se težine nastoje odrediti tako da unaprijed definirana funkcija gubitka bude minimalna na zadanom skupu podataka za učenje. Modifikacija težina obavlja se gradijentnim spustom, pri čemu algoritam povratne propagacije pogreške definira način na koji se određuju gradijenti neurona u unutarnjim slojevima mreže [24]. U nastavku se izvode jednadžbe osvježavanja težina za slučaj kvadratne funkcije gubitka. Analogan izvod može se provesti i za bilo koju derivabilnu funkciju gubitka.

Neka je zadan skup primjera za učenje $D = \{(\boldsymbol{x}_i, \boldsymbol{t}_i) | i = 1,2, \dots, N\}$. $\boldsymbol{x}_i$ je i-ti ulazni d-dimenzionalni uzorak, dok je $\boldsymbol{t}_i$ željeni m-dimenzionalni odziv. Izlaz mreže za ulazni uzorak $\boldsymbol{x}_i$ označen je s $\boldsymbol{y}_i$. Radi jednostavnosti, razmatramo unaprijednu višeslojnu mrežu (slika 9) čiji neuroni koriste sigmoidalnu aktivacijsku funkciju (49). Analogno razmatranje vrijedi i za druge aktivacijske funkcije. Radi lakšeg razumijevanja izvoda koristi se sljedeća notacija s indeksima: $y_{ij}^{(k)}$ predstavlja izlaz j-tog neurona u k-tom sloju za ulazni uzorak $\boldsymbol{x}_i$. Analogno x_{ij} predstavlja j-tu komponentu i-tog ulaznog uzorka. Neka je izlazni sloj označen s $k + 1$. U tom slučaju zadnji skriveni sloj označen je s k. Neka $w_{ij}^{(k)}$ predstavlja težinu sinapse koja povezuje i-ti neuron u k-tom sloju s j-tim neuronom u $(k + 1)$-om sloju.

Jednadžbe osvježavanja težina (61) zahtijevaju određivanje gradijenata funkcije gubitka s obzirom na promatranu težinu. Pritom η predstavlja malu pozitivnu konstantu koja određuje brzinu promjene težina, te se naziva faktorom učenja.

$$w_{ij} \leftarrow w_{ij} - \eta \frac{\partial \mathrm{E}}{\partial w_{ij}} \tag{61}$$

Srednja kvadratna funkcija gubitka definirana je kao:

$$E = \frac{1}{2N}\sum_{s=1}^{N}\sum_{o=1}^{m}\left(t_{so} - y_{so}^{(k+1)}\right)^2 \tag{62}$$

Razmotrimo najprije izračun gradijenta za slučaj izlaznog neurona:

$$\begin{aligned}\frac{\partial \mathrm{E}}{\partial w_{ij}^{(k)}} &= \frac{\partial}{\partial w_{ij}^{(k)}}\left[\frac{1}{2N}\sum_{s=1}^{N}\sum_{o=1}^{m}\left(t_{so} - y_{so}^{(k+1)}\right)^2\right] = \frac{1}{2N}\sum_{s=1}^{N} 2\left(t_{sj} - y_{sj}^{(k+1)}\right)(-1)\frac{\partial y_{sj}^{(k+1)}}{\partial w_{ij}^{(k)}} \\ &= -\frac{1}{N}\sum_{s=1}^{N}\left(t_{sj} - y_{sj}^{(k+1)}\right)\frac{\partial}{\partial w_{ij}^{(k)}}\left[f\left(\sum_{l} w_{lj}^{(k)} y_{sl}^{(k)}\right)\right] \\ &= -\frac{1}{N}\sum_{s=1}^{N}\left(t_{sj} - y_{sj}^{(k+1)}\right)y_{sj}^{(k+1)}\left(1 - y_{sj}^{(k+1)}\right)\frac{\partial}{\partial w_{ij}^{(k)}}\left[\sum_{l} w_{lj}^{(k)} y_{sl}^{(k)}\right] = \\ &= -\frac{1}{N}\sum_{s=1}^{N}\left(t_{sj} - y_{sj}^{(k+1)}\right)y_{sj}^{(k+1)}\left(1 - y_{sj}^{(k+1)}\right)y_{si}^{(k)} = -\frac{1}{N}\sum_{s=1}^{N}\delta_{sj}^{(k+1)}\, y_{si}^{(k)}\end{aligned} \tag{63}$$

Pritom je uvedena pokrata (64), gdje $\delta_{sj}^{(k+1)}$ predstavlja derivaciju funkcije gubitka po aktivaciji j-tog izlaznog neurona za s-ti ulazni uzorak za učenje.

$$\delta_{sj}^{(k+1)} = \left(t_{sj} - y_{sj}^{(k+1)}\right)y_{sj}^{(k+1)}\left(1 - y_{sj}^{(k+1)}\right) \tag{64}$$

Ukoliko je potrebno modificirati težinu nekog od neurona u skrivenim slojevima, gradijent se računa na sljedeći način:

$$\frac{\partial \mathrm{E}}{\partial w_{ij}^{(k-1)}} = \frac{\partial}{\partial w_{ij}^{(k-1)}}\left[\frac{1}{2N}\sum_{s=1}^{N}\sum_{o=1}^{m}\left(t_{so}-y_{so}^{(k+1)}\right)^2\right]$$
$$= \frac{1}{2N}\sum_{s=1}^{N}\sum_{o=1}^{m}2\left(t_{so}-y_{so}^{(k+1)}\right)(-1)\frac{\partial y_{so}^{(k+1)}}{\partial w_{ij}^{(k-1)}}$$
$$= -\frac{1}{N}\sum_{s=1}^{N}\sum_{o=1}^{m}\left(t_{so}-y_{so}^{(k+1)}\right)\frac{\partial}{\partial w_{ij}^{(k-1)}}\left[f\left(\sum_{l} w_{lo}^{(k)} y_{sl}^{(k)}\right)\right]$$
$$= -\frac{1}{N}\sum_{s=1}^{N}\sum_{o=1}^{m}\left(t_{so}-y_{so}^{(k+1)}\right)y_{so}^{(k+1)}\left(1-y_{so}^{(k+1)}\right)\frac{\partial}{\partial w_{ij}^{(k-1)}}\left(\sum_{l} w_{lo}^{(k)} y_{sl}^{(k)}\right) \quad (65)$$
$$= -\frac{1}{N}\sum_{s=1}^{N}\sum_{o=1}^{m}\left(t_{so}-y_{so}^{(k+1)}\right)y_{so}^{(k+1)}\left(1-y_{so}^{(k+1)}\right)w_{jo}^{(k)}\frac{\partial y_{sj}^{(k)}}{\partial w_{ij}^{(k-1)}}$$
$$= -\frac{1}{N}\sum_{s=1}^{N}\frac{\partial y_{sj}^{(k)}}{\partial w_{ij}^{(k-1)}}\sum_{o=1}^{m}\delta_{so}^{(k+1)}w_{jo}^{(k)}$$
$$= -\frac{1}{N}\sum_{s=1}^{N} y_{sj}^{(k)}\left(1-y_{sj}^{(k)}\right)\frac{\partial}{\partial w_{ij}^{(k-1)}}\left(\sum_{l} w_{lj}^{(k-1)} y_{sl}^{(k-1)}\right)\sum_{o=1}^{m}\delta_{so}^{(k+1)}w_{jo}^{(k)}$$
$$= -\frac{1}{N}\sum_{s=1}^{N} y_{sj}^{(k)}\left(1-y_{sj}^{(k)}\right)y_{si}^{(k-1)}\sum_{o=1}^{m}\delta_{so}^{(k+1)}w_{jo}^{(k)} = -\frac{1}{N}\sum_{s=1}^{N} y_{si}^{(k-1)}\delta_{sj}^{(k)}$$

U ovom slučaju je pogreška skrivenog neurona j u sloju k za s-ti ulazni uzorak definirana kao:

$$\delta_{sj}^{(k)} = y_{sj}^{(k)}\left(1-y_{sj}^{(k)}\right)\sum_{o=1}^{m}\delta_{so}^{(k+1)}w_{jo}^{(k)} \quad (66)$$

Vidljivo je kako se pogreška skrivenog neurona j određuje na temelju otežanih pogrešaka onih neurona u narednom sloju na koje je spojen izlaz j-tog neurona.

Konačne jednadžbe osvježavanja težina definirane su na sljedeći način:

$$w_{ij} \leftarrow w_{ij} + \frac{\eta}{N}\sum_{s=1}^{N} y_{si}\delta_{sj} \quad (67)$$

Pritom se lokalni gradijenti δ_{sj} računaju prema (64) odnosno (66) ovisno o tome radi li se o izlaznom ili o neuronu u skrivenom sloju. U praksi se najčešće koristi stohastička varijanta gradijentnog spusta u kojoj se gradijent aproksimira na temelju jednog (*engl. stochastic gradient descent*) ili nekolicine uzoraka (*engl. mini batch gradient descent*). U slučaju stohastičkog gradijentnog spusta jednadžba osvježavanja težina glasi:

$$w_{ij} \leftarrow w_{ij} + \eta y_{si}\delta_{sj} \quad (68)$$

Konačni pseudokod algoritma povratne propagacije pogreške uz primjenu stohastičkog gradijentnog spusta prikazan je u tablici 3.

Tablica 3 Stohastički gradijentni spust - algoritam povratne propagacije pogreške

Postavke:

- $Dest(j)$ – skup neurona iz narednog sloja na koje je spojen izlaz j-tog neurona iz prethodnog sloja
- η - faktor učenja
- t – indeks iteracije algoritma
- $\boldsymbol{w}^{(t)}$ – težine u iteraciji t

1. Nasumično inicijaliziraj težine $\boldsymbol{w}^{(0)}$
2. $t \leftarrow 0$
3. Ponavljaj do konvergencije
   a. $t \leftarrow t + 1$
   b. Izračunaj odziv mreže $\boldsymbol{y}_t$ za ulaz $\boldsymbol{x}_t$
   c. Korigiraj težine $w_{ij}^{(t)} \leftarrow w_{ij}^{(t-1)} + \eta y_{ti}\delta_{tj}$
      i. Za skriveni neuron: $\delta_{tj} = y_{tj}(1 - y_{tj})\sum_{o \in Dest(j)} \delta_{to} w_{jo}$
      ii. Za izlazni neuron: $\delta_{tj} = (t_{tj} - y_{tj})y_{tj}(1 - y_{tj})$

4.2.3.2. Adam

Adam [23] (*engl. adaptive moment estimation*) je inačica stohastičkog gradijentnog spusta koja se često u praksi koristi budući da ubrzava konvergenciju postupka učenja neuronskih mreža. Tijekom učenja *Adam* održava eksponencijalni pomični prosjek gradijenata m_t (69) i njihovih kvadrata v_t (70). Indeks iteracije učenja označen je s t, dok je s g_t označen gradijent u t-toj iteraciji. β_1 i β_2 su konstante otežavanja.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{69}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{70}$$

m_t je procjena prvog momenta (srednje vrijednosti), a v_t drugog momenta (necentrirane varijance), odakle potječe naziv metode. Autori [23] inicijaliziraju početne vrijednosti procjena momenata m_t i v_t na nulu, uslijed čega dolazi do razlike između očekivane vrijednosti procjena i stvarnih momenata. U nastavku (71) je prikazana opisana razlika za slučaj prvog momenta. Analogni izvod vrijedi i za moment drugog reda.

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$

$$m_0 = 0$$

$$m_t = (1-\beta_1) \sum_{i=1}^{t} \beta_1^{t-i} g_i$$

$$E[m_t] = E\left[(1-\beta_1) \sum_{i=1}^{t} \beta_1^{t-i} g_i\right] = \begin{Bmatrix} stacionarnost \\ stohastičkog\ procesa \\ g \end{Bmatrix} \tag{71}$$

$$= (1-\beta_1) \sum_{i=1}^{t} \beta_1^{t-i} E[g_t] = (1-\beta_1^t) E[g_t]$$

Spomenuto neslaganje ispravlja se sljedećom modifikacijom procjena momenata:

$$\widehat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{72}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{73}$$

Adam koristi sljedeće pravilo osvježavanja parametra θ:

$$\theta_t \leftarrow \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \widehat{m}_t \tag{74}$$

Parametar u iteraciji t označen je s θ_t. ϵ je malena pozitivna konstanta koja se koristi za izbjegavanje dijeljenja s nulom. Procjena momenta drugog reda koristi se za prilagodbu stope učenja η, dok se moment prvog reda koristi za ugađanje veličine i smjera pomaka uzimajući u obzir prethodne gradijente. Na taj se način smanjuje utjecaj onih komponenata vektora parametara $\boldsymbol{\theta}$, čiji gradijenti često mijenjaju smjer, čime se postiže brža i usmjerenija pretraga prostora parametara.

Tablica 4 prikazuje pseudokod algoritma *Adam*, prilagođen učenju neuronskih mreža. Radi jednostavnijeg zapisa, iz pseudokoda su izostavljene oznake slojeva.

Tablica 4 Algoritam Adam

Postavke:

- η – faktor učenja
- β_1, β_2 – parametri pomičnog prosjeka
- $\boldsymbol{w}^{(t)}$ – težine u iteraciji t
- ϵ – mala pozitivna konstanta

1. Nasumično inicijaliziraj težine $\boldsymbol{w}^{(0)}$
2. Inicijaliziraj pomične prosjeke $m_{0,j} \leftarrow 0,\ v_{0,j} \leftarrow 0$
3. $t \leftarrow 0$
4. Ponavljaj do konvergencije:
   a. $t \leftarrow t + 1$
   b. Izračunaj odziv mreže $\boldsymbol{y}_t$ za ulaz $\boldsymbol{x}_t$
   c. Za svaki neuron j u svakom sloju odredi pripadni lokalni gradijent δ_{tj} prema izrazima (64), (66)
   d. $m_{tj} \leftarrow \beta_1 m_{t-1,j} + (1-\beta_1)\delta_{tj}$
   e. $v_{t,j} \leftarrow \beta_2 v_{t-1,j} + (1-\beta_2)\delta_{tj}^2$
   f. $\widehat{m}_{t,j} \leftarrow \frac{m_{t,j}}{1-\beta_1^t}$
   g. $\widehat{v}_{t,j} \leftarrow \frac{v_{t,j}}{1-\beta_2^t}$
   h. $w_{ij}^{(t)} \leftarrow w_{ij}^{(t-1)} - \frac{\eta \widehat{m}_{t,j}}{\sqrt{\widehat{v}_{t,j}+\epsilon}}$

4.2.3.3. Funkcije gubitka za klasifikaciju

Kao što je već rečeno u prethodnim poglavljima, za učenje neuronskih mreža koriste se gradijentne metode koje nastoje optimizirati ciljnu funkciju. Spomenuta ciljna funkcija obično se naziva funkcija gubitka, te se tijekom učenja nastoji minimizirati. Kod klasifikacijskih problema, ulaz u klasifikator je slika, a izlaz oznaka razreda kojem ona pripada. Semantička segmentacija može se promatrati kao višestruka klasifikacija, gdje se svakom pikselu želi pridružiti oznaka njegovog razreda. Postoji niz funkcija gubitka, od kojih su neke istaknute u nastavku.

Najjednostavnija funkcija gubitka je nula-jedan gubitak. Ideja ove funkcije jest kazniti pogrešno klasificirane primjere s gubitkom iznosa jedan, dok se ispravno klasificirani primjeri ne kažnjavaju:

$$loss_{01} = \sum_{i=1}^{|D|} 1\{y_i \neq t_i\} \tag{75}$$

$D = \{(\boldsymbol{x}_i, t_i)\}$ je skup primjera za učenje, t_i predstavlja željeni, a y_i dobiveni izlaz modela. $1\{\cdot\}$ je indikatorska funkcija koja poprima vrijednost 1 kada joj je argument istinit. Glavni nedostatak funkcije gubitka nula-jedan je njezina nederivabilnost. Zbog toga se za gradijentne metode učenja često primjenjuje negativna log izglednost:

$$\begin{aligned} loss_{nll} &= -\ln \mathcal{L}(\boldsymbol{\theta}|D) = -\ln P(D|\boldsymbol{\theta}) \\ &= -\ln \prod_{i=1}^{|D|} P(t_i|x_i, \boldsymbol{\theta}) = -\sum_{i=1}^{|D|} \ln P(t_i|x_i, \boldsymbol{\theta}) \end{aligned} \tag{76}$$

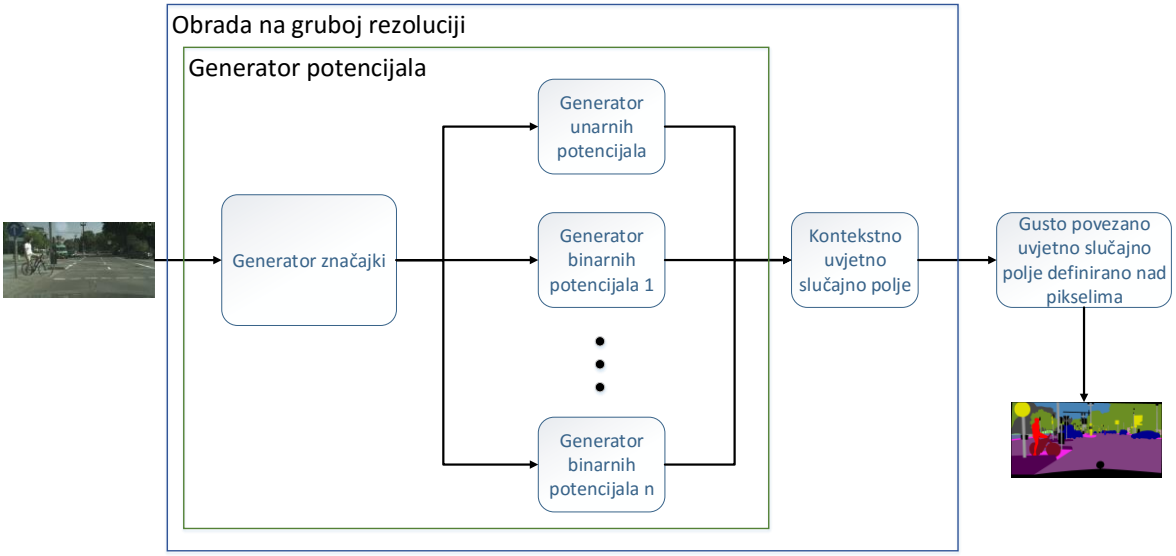
Pritom je s $\boldsymbol{\theta}$ označen skup parametara modela. Opisana funkcija gubitka bit će minimalna ukoliko su parametri modela odabrani tako da primjeri dostupni u skupu za učenje budu što vjerojatniji. Ukoliko se radi o binarnoj klasifikaciji, funkcija gubitka (76) može prikazati kao:

$$\begin{aligned} loss_{nll} &= -\sum_{i=1}^{|D|} \ln P(t_i|x_i, \boldsymbol{\theta}) = -\sum_{i=1}^{|D|} \ln\left[y_i^{t_i}\left(1 - y_i^{1-t_i}\right)\right] \\ &= -\sum_{i=1}^{|D|} \{t_i \ln y_i + (1 - t_i)\ln(1 - y_i)\} \end{aligned} \tag{77}$$

Ovako definirana funkcija gubitka naziva se još i pogreška unakrsne entropije.

5. Implementacija

U ovom poglavlju opisuju se detalji implementiranog modela za semantičku segmentaciju. Model se sastoji od dva dijela: obrade slike na gruboj rezoluciji (generator potencijala i kontekstno uvjetno slučajno polje) i uvjetnog slučajnog polja definiranog nad pikselima. Shema implementiranog sustava prikazana je na slici 12.



Slika 12 Shema implementiranog sustava za semantičku segmentaciju

U nastavku se detaljnije opisuju njihove arhitekture, kao i pripadni postupci učenja i zaključivanja.

5.1. Obrada na gruboj rezoluciji

Obrada na gruboj rezoluciji predstavlja početni korak za semantičku segmentaciju u implementiranom sustavu. Sastoji se od duboke konvolucijske neuronske mreže i kontekstnog uvjetnog slučajnog polja. Zadatak implementirane konvolucijske mreže jest generiranje unarnih i binarnih potencijala, dok se uvjetno slučajno polje koristi kako bi se potencijali kombinirali u vjerojatnosnu distribuciju. Implementirani model temelji se na radu Lina i ostalih [6].

5.1.1. Arhitektura konvolucijske neuronske mreže

Arhitektura implementirane neuronske mreže može se podijeliti na tri osnovna dijela:

- Generator značajki
- Generator unarnih potencijala

- Generatori binarnih potencijala

Generator značajki je konvolucijska neuronska mreža čiji je zadatak za svaki piksel na temelju ulazne slike generirati vektor značajki koji ga opisuje. Generator je konvolucijska neuronska mreža čiji početni dio odgovara mreži VGG16 [7]. Tablica 5 prikazuje konkretnu arhitekturu implementirane konvolucijske mreže.

Tablica 5 Arhitektura generatora značajki

Naziv sloja	Veličina jezgre / pomak	Broj izlaznih mapa	Aktivacijska funkcija	Grupna normalizacija
1. blok				
conv1_1	3x3 / 1	64	ReLU	NE
conv1_2	3x3 / 1	64	ReLU	NE
pool1	2x2 / 2	64	/	NE
2. blok				
conv2_1	3x3 / 1	128	ReLU	NE
conv2_2	3x3 / 1	128	ReLU	NE
pool2	2x2 / 2	128	/	NE
3. blok				
conv3_1	3x3 / 1	256	ReLU	NE
conv3_2	3x3 / 1	256	ReLU	NE
conv3_3	3x3 / 1	256	ReLU	NE
pool3	2x2 / 2	256	/	NE
4. blok				
conv4_1	3x3 / 1	512	ReLU	NE
conv4_2	3x3 / 1	512	ReLU	NE
conv4_3	3x3 / 1	512	ReLU	NE
pool4	2x2 / 2	512	/	NE
5. blok				
conv5_1	3x3 / 1	512	ReLU	NE
conv5_2	3x3 / 1	512	ReLU	NE
conv5_3	3x3 / 1	512	ReLU	NE
pool5	2x2 / 1	512	/	NE
6. blok				
conv6_1	1x1 / 1	1024	ReLU	DA
conv6_2	3x3 / 1	512	ReLU	DA
conv6_3	3x3 / 1	512	ReLU	DA

Arhitektura generatora značajki može se podijeliti u šest blokova pri čemu svaki blok započinje s konvolucijskim, te završava sa sažimajućim slojem (izuzev zadnjeg bloka). Prva četiri bloka završavaju sa sažimajućim slojevima koji smanjuju dimenzije ulaza dva puta, dok peti sažimajući sloj ne mijenja dimenzije svog ulaza.

Prvih pet blokova odgovara mreži VGG16, izuzev zadnjeg sažimajućeg sloja koji u implementiranom modelu koristi jedinični korak zbog čega ne mijenja dimenziju svog ulaza. Svi sažimajući slojevi vrše sažimanje odabirom maksimalnog odziva (*engl. max pooling*). Šesti konvolucijski blok sadrži tri konvolucijska sloja, te izlaz posljednjeg konvolucijskog sloja predstavlja mapu značajki. Svi konvolucijski slojevi šestog bloka koriste grupnu (*engl. batch*) normalizaciju. Na izlaze svih konvolucijskih slojeva primjenjuje se ReLU aktivacijska funkcija. Konvolucijski slojevi ne mijenjaju dimenzije svojih ulaza budući da se prije obavljanja konvolucije rubovi ulaza proširuju (*engl. padding*) nulama. Budući da mreža efektivno sadrži četiri sažimajuća sloja, koji smanjuju dimenzije svojih ulaza dva puta, dimenzije ulazne slike će na izlazu generatora značajki biti reducirane 16 puta. U takvoj reduciranoj slici svaki piksel je opisan s jednim vektorom značajki iz prethodno spomenute mape značajki.

Druga komponenta generatora potencijala jest generator unarnih potencijala. On se sastoji od dva konvolucijska sloja čije su jezgre dimenzija $1x1$. Arhitektura generatora unarnih potencijala prikazana je u tablici 6.

Tablica 6 Arhitektura generatora unarnih (gore) i binarnih (dolje) potencijala

Naziv sloja	Veličina jezgre / pomak	Broj izlaznih mapa	Aktivacijska funkcija	Grupna normalizacija
Generator unarnih potencijala				
convu_1	1x1 / 1	512	ReLU	DA
convu_2	1x1 / 1	K	/	NE

Generator binarnih potencijala				
convb_1	1x1 / 1	512	ReLU	DA
convb_2	1x1 / 1	K^2	/	NE

Broj izlaznih mapa K zadnjeg konvolucijskog sloja generatora unarnih potencijala odgovara broju semantičkih razreda skupa slika. Ulaz u generator unarnih potencijala jest mapa značajki koju generira generator značajki. Generator unarnih potencijala za svaki piksel u ulaznoj mapi značajki generira onoliko izlaznih vrijednosti koliko postoji semantičkih razreda. Dobivene vrijednosti predstavljaju sigurnost mreže da se pripadni piksel treba klasificirati u odgovarajući semantički razred.

Zadnja komponenta generatora potencijala jesu generatori binarnih potencijala. Arhitektura generatora binarnih potencijala također je prikazana u tablici 6. Razlikuje se u odnosu na generator unarnih potencijala po dimenzijama ulaza, te broju izlaznih mapa zadnjeg konvolucijskog sloja. Budući da opisani dio modela treba generirati binarne potencijale koji modeliraju prikladnost semantičkih razreda dvaju piksela, ulaz u generator binarnih potencijala čini modificirana mapa značajki koja se dobije tako da se za svaki par piksela (u reduciranoj slici) koji se nalaze u unaprijed definiranom susjedstvu spoje pripadni vektori značajki iz mape značajki koju generira generator značajki. Broj izlaznih mapa zadnjeg konvolucijskog sloja odgovara kvadratu broja semantičkih razreda, budući da je potrebno modelirati prikladnost svih mogućih kombinacija semantičkih oznaka dvaju piksela.

Na izlaze posljednjih konvolucijskih slojeva generatora unarnih i binarnih potencijala ne primjenjuje se eksplicitno softmax aktivacijska funkcija kako bi nepromijenjeni izlazi mreže (sigurnost u klasifikaciju) bili dostupni metodi srednjeg polja kod zaključivanja. Softmax se implicitno primjenjuje prilikom izračuna funkcije gubitka. Za više detalja pogledati poglavlje 5.1.4.

5.1.2. Unarni i binarni potencijali

U prethodnom poglavlju spomenute su komponente modela koje generiraju unarne i binarne potencijale. Kako se izlazi generatora unarnih, odnosno binarnih potencijala tumače kao mjere sigurnosti, pripadni potencijali definiraju se na sljedeći način:

$$U(y_p, \boldsymbol{x}_p|\boldsymbol{\theta}_U) = -z_{p,y_p}(\boldsymbol{x}|\boldsymbol{\theta}_U) \tag{78}$$

$$V(y_p, y_q, \boldsymbol{x}_{pq}|\boldsymbol{\theta}_V) = -z_{p,q,y_p,y_q}(\boldsymbol{x}|\boldsymbol{\theta}_V) \tag{79}$$

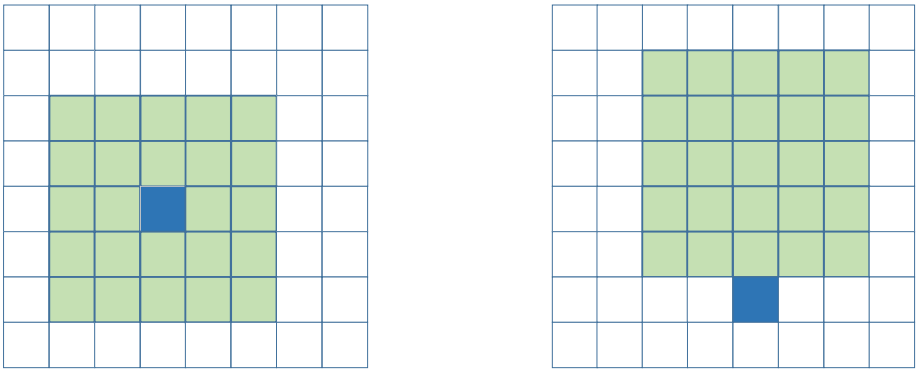
Pritom U i V predstavljaju unarne odnosno binarne potencijale, p i q indekse piksela u reduciranoj slici, y_p semantički razred piksela p. Parametri generatora unarnih potencijala označeni su s $\boldsymbol{\theta}_U$ dok su parametri generatora binarnih potencijala označeni s $\boldsymbol{\theta}_V$. Mapa značajki na izlazu generatora značajki označena ja s $\boldsymbol{x}$, dok $\boldsymbol{x}_p$ odnosno $\boldsymbol{x}_{pq}$ predstavlja vektor značajki koji opisuje piksel p, odnosno par piksela (p, q). z_{p,y_p} predstavlja mjeru sigurnosti (izlaz) generatora unarnih potencijala koja odgovara pikselu p i semantičkom razredu y_p. Analogno z_{p,q,y_p,y_q} predstavlja mjeru

sigurnosti generatora binarnih potencijala da par piksela (p, q) treba biti klasificiran u semantičke razrede (y_p, y_q).

U implementiranom modelu unarni i binarni potencijali se uče, te se generiraju pomoću konvolucijske neuronske mreže. Unarni potencijali modeliraju cijenu klasificiranja pojedinog piksela u reduciranoj slici (promatranog u izoliranosti) u određeni semantički razred. Ekvivalentno binarni potencijali modeliraju cijenu klasificiranja para piksela. Pritom se ne promatraju svi mogući pikseli nego se definiraju susjedstva, te se promatraju kombinacije samo onih piksela koji se nalaze u definiranom susjedstvu. Binarnim potencijalima modelira se cijena supojavljivanja semantičkih oznaka piksela u definiranom susjedstvu. U okviru ovog rada promatraju se dva tipa susjedstava:

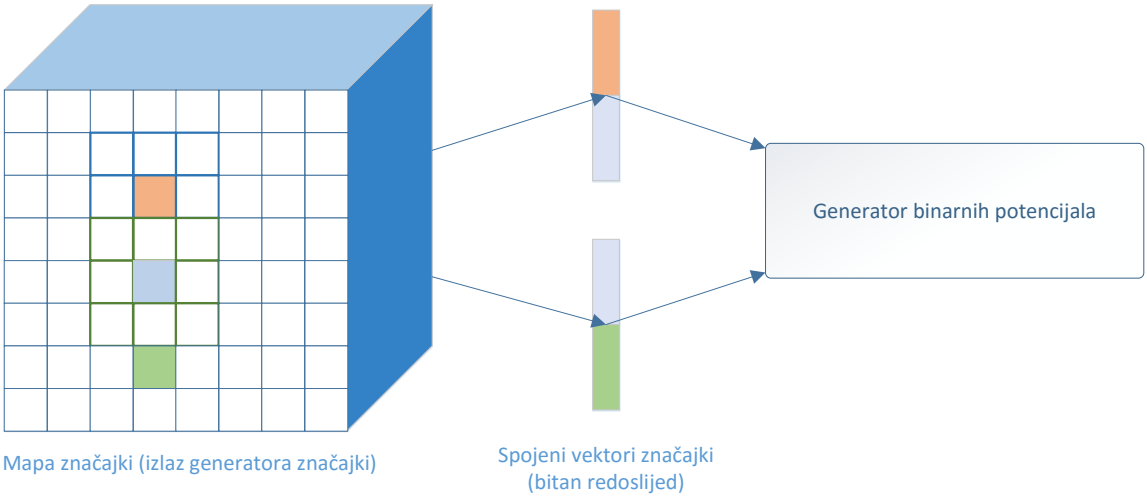
- „okruženje“
- „iznad/ispod“

Navedena susjedstva prikazuje slika 13. Pritom su veličine susjedstava ograničene kako bi zaključivanje pomoću uvjetnog slučajnog polja bilo efikasno.



Slika 13 Dvije vrste susjedstava: okruženje (lijevo) i iznad/ispod (desno)

U obje vrste susjedstava promatraju se odnosi između plavog piksela i ostalih piksela u zelenom kvadratu (slika 13). Svakoj kombinaciji odgovara jedan binarni potencijal $V(Y_p, Y_q, \boldsymbol{x}_{pq}|\boldsymbol{\theta}_V)$. Pritom binarni potencijali mogu biti asimetrične funkcije, odnosno $V(y_p, y_q, \boldsymbol{x}_{pq}) \neq V(y_q, y_p, \boldsymbol{x}_{qp})$. Drugim riječima, redoslijed ulaznih vektora značajki piksela p i q definira njihov prostorni raspored u slici. U slučaju relacije „iznad/ispod“, binarni potencijal $V(y_p, y_q, \boldsymbol{x}_{pq})$ modelira situaciju u kojoj se piksel q nalazi iznad piksela p. Promotrimo što to točno znači u okviru neuronske mreže koja generira binarne potencijale (slika 14).



Slika 14 Asimetričnost binarnih potencijala

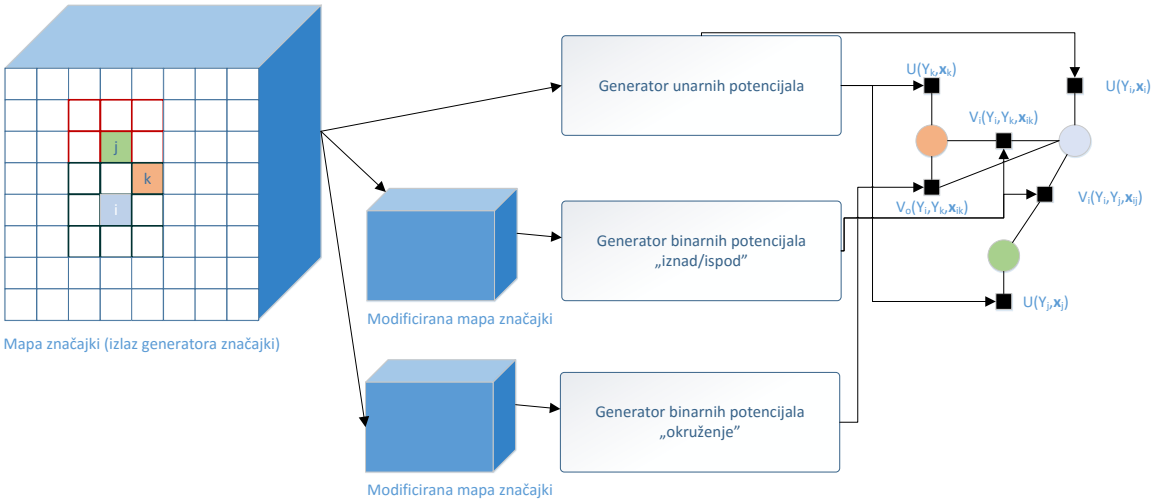
Kao što je već rečeno u prethodnom poglavlju, ulaz u generator binarnih potencijala dobije se spajanjem vektora značajki susjednih piksela (slika 13) iz izvorne mape značajki. Zbog toga je dubina mape značajki (treća dimenzija) za binarne potencijale duplo veća u odnosu na izlaz generatora značajki. Slika 14 ne prikazuje čitavu modificiranu mapu značajki nego spojene vektore značajki za dva piksela promatrajući susjedstvo „iznad/ispod“. Narančasti piksel nalazi se u susjedstvu plavog piksela te se njihovi vektori značajki spajaju na način da vektor za plavi piksel bude ispod vektora narančastog piksela. Analogno, plavi piksel nalazi se u susjedstvu zelenog, te prilikom spajanja njihovih vektora značajki, vektor plavog piksela bit će smješten iznad vektora zelenog. Na ovaj način se omogućava mreži koja generira binarne potencijale razlikovanje prostorne pozicije (položaja) piksela.

Bitno je naglasiti kako svako definirano susjedstvo predstavlja zasebnu mrežu (generator binarnih potencijala) u predstavljenom modelu. Dakle u slučaju dvaju susjedstava, postoje dva generatora binarnih potencijala.

5.1.3. Kontekstno uvjetno slučajno polje

Za potrebe kombiniranja unarnih i binarnih potencijala u procesu zaključivanja koristi se uvjetno slučajno polje definirano nad varijablama $\boldsymbol{Y} \cup \boldsymbol{X}$. Uvjetno slučajno polje nazivamo kontekstnim budući da za svaki piksel promatra kontekst (semantičke razrede) piksela u njegovom susjedstvu. Varijabla X_i u $\boldsymbol{X}$ odgovara vektoru značajki piksela i u reduciranoj slici. Navedeni vektor značajki sastavni je dio mape značajki koju generira generator značajki. Svaka slučajna varijabla $Y_i \in \boldsymbol{Y}$ modelira

semantički razred u koji je klasificiran piksel i u reduciranoj slici. Isječak opisanog uvjetnog slučajnog polja i njegovu vezu s mapom značajki prikazuje slika 15.



Slika 15 Uvjetno slučajno polje

Radi jednostavnijeg prikaza uvjetnog slučajnog polja izostavljeni su čvorovi koji odgovaraju varijablama $\boldsymbol{X}$. Prikazani su samo binarni potencijali piksela i. Kako se pikseli j i k nalaze iznad piksela i, pripadni čvorovi će biti povezani preko faktora koji predstavljaju binarne potencijale V_i. Piksel k nalazi se i u susjedstvu „okruženje" piksela i zbog čega graf sadrži faktor V_o između piksela i i k. Opisano uvjetno slučajno polje modelira sljedeću distribuciju:

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \frac{1}{Z(\boldsymbol{X})} e^{-E(\boldsymbol{Y},\boldsymbol{X})} \tag{80}$$

Z predstavlja normalizacijsku konstantu (81), dok je E energijska funkcija koja modelira kompatibilnost ulazne slike $\boldsymbol{x}$ i predikcije $\boldsymbol{y}$. Pod predikcijom se podrazumijeva semantički razred pridijeljen svakom pikselu u $\boldsymbol{x}$.

$$Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}} e^{-E(\boldsymbol{y},\boldsymbol{x})} \tag{81}$$

Energija za ulaznu sliku $\boldsymbol{x}$ i predikciju $\boldsymbol{y}$ definirana je kao:

$$E(\boldsymbol{y},\boldsymbol{x}) = \sum_{p \in N_U} U(y_p, \boldsymbol{x}_p) + \sum_{p \in N_U, q \in S_o(p)} V_o(y_p, y_q, \boldsymbol{x}_{pq}) + \sum_{p \in N_U, q \in S_i(p)} V_i(y_p, y_q, \boldsymbol{x}_{pq}) \tag{82}$$

Radi konciznijeg zapisa u prethodnim izrazima izostavljena je oznaka parametara $\boldsymbol{\theta}$ iz definicije unarnih odnosno binarnih potencijala. N_U predstavlja skup čvorova

uvjetnog slučajnog polja, $S_o(p)$ skup susjeda čvora p s obzirom na relaciju „okruženja“, a $S_i(p)$ skup susjeda čvora p s obzirom na relaciju „iznad/ispod“. V_o i V_i su binarni potencijali za relaciju „okruženje“ odnosno „iznad/ispod“.

5.1.4. Učenje

Učenje generatora potencijala svodi se na učenje parametara cjelokupne mreže (generatora značajki, unarnih i binarnih potencijala). Radi jednostavnosti zapisa svi parametri modela označeni su s $\boldsymbol{\theta}$.

Tipičan pristup učenju uvjetnog slučajnog polja (odnosno parametara modela) svodi se na minimizaciju negativne log izglednosti, koja se može zapisati kao:

$$-\sum_{i=1}^{|D|} \ln P(\boldsymbol{y}^i \,|\boldsymbol{x}^i, \boldsymbol{\theta}) = \sum_{i=1}^{|D|} \left[E(\boldsymbol{y}^i, \boldsymbol{x}^i, \boldsymbol{\theta}) + \ln Z(\boldsymbol{x}^i, \boldsymbol{\theta})\right] \tag{83}$$

Dodavanjem regularizacijskog člana u izraz (83) dobiva se sljedeća funkcija gubitka:

$$\frac{\lambda}{2}\|\theta\|_2^2 + \sum_{i=1}^{|D|} \left[E(\boldsymbol{y}^i, \boldsymbol{x}^i, \boldsymbol{\theta}) + \ln Z(\boldsymbol{x}^i, \boldsymbol{\theta})\right] \tag{84}$$

U prethodnim izrazima D predstavlja skup primjera za učenje. Za učenje parametara $\boldsymbol{\theta}$ mogao bi se primijeniti gradijentni spust. Gradijent energijske funkcije može se odrediti primjenom lančanog pravila budući da se radi o neuronskoj mreži. Problem međutim predstavlja gradijent logaritma particijske funkcije (85), čiji je izračun računski prezahtjevan zbog zbroja po svim mogućim mapama oznaka $\boldsymbol{y}$. Ako postoji 20 semantičkih razreda i 10^6 piksela u slici, bilo bi potrebno obaviti zbroj po 20^{10^6} mogućih realizacija $\boldsymbol{y}$, što je računski prezahtjevno.

$$\nabla_{\boldsymbol{\theta}} \ln Z(\boldsymbol{x}, \boldsymbol{\theta}) = -\sum_{\boldsymbol{y}} \frac{e^{-E(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta})}}{\sum_{\boldsymbol{y}'} e^{-E(\boldsymbol{y}', \boldsymbol{x}, \boldsymbol{\theta})}} \nabla_{\boldsymbol{\theta}} \left[E(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})\right] \tag{85}$$

U implementiranom modelu, problem je riješen primjenom aproksimativne metode učenja uvjetnog slučajnog polja. Uvjetno slučajno polje uči se po dijelovima, na način da se funkcija izglednosti formulira preko izglednosti definiranim na potencijalima:

$$
\begin{aligned}
&P(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{x})} e^{-E(\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta})} \\
&= \frac{1}{Z(\boldsymbol{x})} \prod_{p \in N_U} e^{-U(y_p,\boldsymbol{x}_p,\boldsymbol{\theta}_U)} \prod_{p \in N_U, q \in S_o(p)} e^{-V_o(y_p,y_q,\boldsymbol{x}_{pq},\boldsymbol{\theta}_{V_o})} \prod_{p \in N_U, q \in S_i(p)} e^{-V_i\left(y_p,y_q,\boldsymbol{x}_{pq},\boldsymbol{\theta}_{V_i}\right)} \\
&= \frac{1}{Z_1(\boldsymbol{x})} \prod_{p \in N_U} e^{-U(y_p,\boldsymbol{x}_p,\boldsymbol{\theta}_U)} \frac{1}{Z_2(\boldsymbol{x})} \prod_{p \in N_U, q \in S_o(p)} e^{-V_o(y_p,y_q,\boldsymbol{x}_{pq},\boldsymbol{\theta}_{V_o})} \frac{1}{Z_3(\boldsymbol{x})} \prod_{p \in N_U, q \in S_i(p)} e^{-V_i\left(y_p,y_q,\boldsymbol{x}_{pq},\boldsymbol{\theta}_{V_i}\right)} \\
&= \prod_{p \in N_U} P_U(y_p|\boldsymbol{x},\boldsymbol{\theta}_U) \prod_{p \in N_U, q \in S_o(p)} P_{V_o}(y_p,y_q|\boldsymbol{x},\boldsymbol{\theta}_{V_o}) \prod_{p \in N_U, q \in S_i(p)} P_{V_i}(y_p,y_q|\boldsymbol{x},\boldsymbol{\theta}_{V_i})
\end{aligned}
\tag{86}
$$

Podjela particijske funkcije Z na Z_1, Z_2 i Z_3 u izrazu (86) je pretpostavka koju navedeni aproksimativni postupak uvodi, te općenito ne vrijedi. Izglednost zasnovana na unarnim potencijalima definirana je izrazom (87), dok su izglednosti zasnovane na binarnim potencijalima definirane izrazom (88).

$$
P_U\left(y_p\middle|\boldsymbol{x},\boldsymbol{\theta}_U\right) = \frac{e^{-U(y_p,\boldsymbol{x}_p,\boldsymbol{\theta}_U)}}{\sum_{y_p'} e^{-U(y_p',\boldsymbol{x}_p,\boldsymbol{\theta}_U)}} \tag{87}
$$

$$
P_{V_k}\left(y_p,y_q\middle|\boldsymbol{x},\boldsymbol{\theta}_{V_k}\right) = \frac{e^{-V_k(y_p,y_q,\boldsymbol{x}_{pq},\boldsymbol{\theta}_{V_k})}}{\sum_{y_p',y_q'} e^{-V_k(y_p',y_q',\boldsymbol{x}_{pq},\boldsymbol{\theta}_{V_k})}}, k \in \{o,i\} \tag{88}
$$

Konačna funkcija gubitka za učenje parametara uvjetnog slučajnog polja po dijelovima definirana je kao:

$$
\begin{aligned}
\frac{\lambda}{2}\|\theta\|_2^2 - \sum_{i=1}^{|D|} \Bigg[&\sum_{p \in N_U} \ln P_U(y_p|\boldsymbol{x}^i,\boldsymbol{\theta}_U) + \sum_{p \in N_U, q \in S_o(p)} \ln P_{V_o}(y_p,y_q|\boldsymbol{x}^i,\boldsymbol{\theta}_{V_o}) \\
&+ \sum_{p \in N_U, q \in S_i(p)} \ln P_{V_i}(y_p,y_q|\boldsymbol{x}^i,\boldsymbol{\theta}_{V_i}) \Bigg]
\end{aligned}
\tag{89}
$$

Prikazana definicija pogodna je za primjenu stohastičkog gradijentnog spusta, budući da ne uključuje globalnu particijsku funkciju Z kao izraz (84). Budući da su izglednosti temeljene na potencijalima definirane izrazima (87) i (88) *softmax* normalizacijske funkcije, njihov je gradijent moguće odrediti, te je učenje parametara uvjetnog slučajnog polja moguće efikasno provesti.

5.1.5. Zaključivanje

Proces zaključivanja započinje prolaskom ulazne slike kroz generator potencijala. Dobiveni unarni i binarni potencijali koriste se u uvjetnom slučajnom polju za zaključivanje. Kako bi se zaključivanje moglo efikasno provesti koristi se aproksimativna metoda srednjeg polja, koja izvornu distribuciju $P(\boldsymbol{Y}|\boldsymbol{X})$ nastoji

aproksimirati distribucijom $Q(\boldsymbol{Y}|\boldsymbol{X}) = \prod_i Q_i(Y_i|\boldsymbol{X})$. Uzimajući u obzir prethodno definirane unarne i binarne potencijale, iterativne jednadžbe općenite metode srednjeg polja (45) prilagođene su implementiranom modelu. Budući da je uvjetno slučajno polje definirano preko energije i potencijala (82), izraz (80) može se prilagoditi kako bi odgovarao općenitoj definiciji (27) temeljenoj na faktorima ϕ.

$$
\begin{aligned}
P(\boldsymbol{Y}|\boldsymbol{X}) &= \frac{1}{Z(\boldsymbol{X})} e^{-E(\boldsymbol{Y},\boldsymbol{X})} \\
&= \frac{1}{Z(\boldsymbol{X})} e^{-\left[\sum_{p\in N_U} U(y_p,\boldsymbol{x}_p)+\sum_{p\in N_U, q\in S_o(p)} V_o(y_p,y_q,\boldsymbol{x}_{pq})+\sum_{p\in N_U, q\in S_i(p)} V_i(y_p,y_q,\boldsymbol{x}_{pq})\right]} \\
&= \frac{1}{Z(\boldsymbol{X})} \prod_{p\in N_U} e^{-U(y_p,\boldsymbol{x}_p)} \prod_{p\in N_U, q\in S_o(p)} e^{-V_o(y_p,y_q,\boldsymbol{x}_{pq})} \prod_{p\in N_U, q\in S_i(p)} e^{-V_i(y_p,y_q,\boldsymbol{x}_{pq})}
\end{aligned}
\tag{90}
$$

Usporedbom izraza (90) i (27) vidljivo je kako faktori ϕ odgovaraju eksponentima negativnih potencijala. Imajući to u vidu, iterativna jednadžba općenite metode srednjeg polja (45) može se raspisati na sljedeći način:

$$
\begin{aligned}
Q_i(y_i|\boldsymbol{x}) &= \frac{1}{Z_i(\boldsymbol{x})} \exp\left\{ \sum_{\phi: Y_i\in \boldsymbol{Y}_\phi} E_{(\boldsymbol{Y}_\phi-\{Y_i\})\sim Q}\left[\ln \phi\left(\boldsymbol{Y}_\phi, y_i|\boldsymbol{x}\right)\right] \right\} \\
&= \frac{1}{Z_i(\boldsymbol{x})} \exp\Bigg\{ E_{(\{Y_i\}-\{Y_i\})\sim Q}[-U(y_i,\boldsymbol{x}_i)] + \sum_{j\in S_o(i)} E_{Y_j\sim Q}\left[-V_o\left(y_i,Y_j,\boldsymbol{x}_{ij}\right)\right] \\
&\qquad + \sum_{k: i\in S_o(k)} E_{Y_k\sim Q}[-V_o(Y_k,y_i,\boldsymbol{x}_{ki})] + \sum_{j\in S_i(i)} E_{Y_j\sim Q}\left[-V_i\left(y_i,Y_j,\boldsymbol{x}_{ij}\right)\right] \\
&\qquad + \sum_{k: i\in S_i(k)} E_{Y_k\sim Q}[-V_i(Y_k,y_i,\boldsymbol{x}_{ki})] \Bigg\} \\
&= \frac{1}{Z_i(\boldsymbol{x})} \exp\Bigg\{ -U(y_i,\boldsymbol{x}_i) - \sum_{j\in S_o(i)} \sum_{y_j\in Val(Y_j)} Q_j\left(y_j|\boldsymbol{x}\right) V_o\left(y_i,y_j,\boldsymbol{x}_{ij}\right) \\
&\qquad - \sum_{k: i\in S_o(k)} \sum_{y_k\in Val(Y_k)} Q_k(y_k|\boldsymbol{x}) V_o(y_k,y_i,\boldsymbol{x}_{ki}) \\
&\qquad - \sum_{j\in S_i(i)} \sum_{y_j\in Val(Y_j)} Q_j\left(y_j|\boldsymbol{x}\right) V_i\left(y_i,y_j,\boldsymbol{x}_{ij}\right) \\
&\qquad - \sum_{k: i\in S_i(k)} \sum_{y_k\in Val(Y_k)} Q_k(y_k|\boldsymbol{x}) V_i(y_k,y_i,\boldsymbol{x}_{ki}) \Bigg\}
\end{aligned}
\tag{91}
$$

Prilagođeni iterativni algoritam srednjeg polja prikazuje tablica 7.

Tablica 7 Prilagođeni algoritam srednjeg polja

1. *Inicijalizacija:* $Q_i(y_i|\boldsymbol{x}) \leftarrow \exp\{-U(y_i, \boldsymbol{x}_i)\}, \forall i \in N_U, \forall y_i \in Val(Y_i)$
   a. *Normaliziraj* $Q_i(Y_i|\boldsymbol{x})$ *za* $\forall i \in N_U$
2. *Ponavljaj zadani broj iteracija:*
   a. *Za svaki* Q_i *ponovi:*
      i. Za svaki $y_i \in Val(Y_i)$:
         1. $Q_i(y_i|\boldsymbol{x}) \leftarrow \exp\Big\{-U\big(y_i, \boldsymbol{x}_i\big) - \sum_{j \in S_o(i)} \sum_{y_j \in Val(Y_j)} Q_j\left(y_j|\boldsymbol{x}\right) V_o\left(y_i, y_j, \boldsymbol{x}_{ij}\right) - \sum_{k:i \in S_o(k)} \sum_{y_k \in Val(Y_k)} Q_k\big(y_k|\boldsymbol{x}\big) V_o\big(y_k, y_i, \boldsymbol{x}_{ki}\big) - \sum_{j \in S_i(i)} \sum_{y_j \in Val(Y_j)} Q_j\left(y_j|\boldsymbol{x}\right) V_i\left(y_i, y_j, \boldsymbol{x}_{ij}\right) - \sum_{k:i \in S_i(k)} \sum_{y_k \in Val(Y_k)} Q_k\big(y_k|\boldsymbol{x}\big) V_i\big(y_k, y_i, \boldsymbol{x}_{ki}\big)\Big\}$
      ii. Normaliziraj $Q_i(Y_i|\boldsymbol{x})$ tako da $\sum_{y_i \in Val(Y_i)} Q_i(y_i|\boldsymbol{x}) = 1$

Metoda srednjeg polja rezultira aproksimacijom $Q(\boldsymbol{Y}|\boldsymbol{X})$ izvorne distribucije $P(\boldsymbol{Y}|\boldsymbol{X})$. Navedena distribucija koristi se za inicijalizaciju unarnih potencijala uvjetnog slučajnog polja definiranog nad pikselima. Ukoliko se želi generirati predikcija za ulaznu sliku u ovom dijelu modela, može se primijeniti MAP procjena nad $Q(\boldsymbol{Y}|\boldsymbol{X})$ (92) uzimajući u obzir zapis distribucije Q kao produkta marginalnih distribucija. Dobivena semantička mapa definirana je na smanjenoj rezoluciji u odnosu na ulaznu sliku.

$$\boldsymbol{y}^* = \underset{\boldsymbol{y}}{\operatorname{argmax}}\, Q(\boldsymbol{y}|\boldsymbol{x}) = \underset{\boldsymbol{y}}{\operatorname{argmax}} \prod_i Q_i(y_i|\boldsymbol{x})$$
$$y_i^* = \underset{y_i}{\operatorname{argmax}}\, Q_i(y_i|\boldsymbol{x}) \qquad (92)$$

5.2. Uvjetno slučajno polje definirano nad pikselima

Za razliku od kontekstnog uvjetnog slučajnog polja, koje je definirano nad konvolucijskim značajkama, u ovom poglavlju opisuje se uvjetno slučajno polje čiji su binarni potencijali definirani nad vektorima boje i pozicije piksela.

Izlaz kontekstnog uvjetnog slučajnog polja je mapa vjerojatnosnih distribucija po semantičkim razredima za svaki piksel reducirane slike. Budući da sustav za semantičku segmentaciju treba na izlazu generirati semantičku mapu za ulaznu sliku izvorne rezolucije, mapa vjerojatnosnih distribucija se bilinearnom interpolacijom proširuje do izvornih dimenzija. Reducirana mapa značajki nad kojom

djeluje kontekstno uvjetno slučajno polje dobivena je uzastopnim sažimanjem izvorne slike. Neizbježna posljedica sažimanja je gubitak informacije zbog čega će semantička mapa izvornih dimenzija sadržavati dosta šuma prvenstveno uz rubove objekata.

Kako bi se smanjio negativni utjecaj proširenja slike bilinearnom interpolacijom koristi se gusto povezano uvjetno slučajno polje definirano nad pikselima [8].

5.2.1. Definicija

Nedostatak kontekstnog uvjetnog slučajnog polja opisanog u poglavlju 5.1.3 jest ograničen doseg binarnih potencijala. Budući da se binarni potencijali uče pomoću neuronske mreže, efikasna implementacija zaključivanja iziskuje ograničenje susjedstva svakog piksela. Unatoč tom ograničenju, susjedstvo nije toliko malo s obzirom na rezoluciju na kojoj se primjenjuje. Navedeni problem ograničenog susjedstva rješava gusto povezano uvjetno slučajno polje definirano nad pikselima, čija specifična definicija binarnih potencijala omogućava efikasno zaključivanje unatoč povezanosti svakog piksela sa svim ostalim pikselima u slici. U nastavku se formalno definira spomenuto uvjetno slučajno polje, te pripadna notacija.

Neka je $\boldsymbol{Y}$ vektor slučajnih varijabli $Y_1 Y_2 Y_3 \dots Y_n$. Svaka slučajna varijabla Y_i opisuje semantički razred u koji je klasificiran piksel i. Skup mogućih vrijednosti (semantičkih razreda) koje može poprimiti varijabla Y_i označen je s $Val(Y_i) = \{l_1, l_2, \dots, l_k\}$, gdje je k ukupan broj semantičkih razreda u skupu slika. Broj piksela u ulaznoj slici označen je s n. Neka je $\boldsymbol{X}$ vektor slučajnih varijabli $X_1 X_2 X_3 \dots X_n$. Svaka slučajna varijabla X_i predstavlja vektor boja piksela i.

Gusto povezano uvjetno slučajno polje definirano nad pikselima (X, Y) opisuje sljedeću distribuciju:

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \frac{1}{Z(\boldsymbol{X})} e^{-E(\boldsymbol{Y}|\boldsymbol{X})} \tag{93}$$

Pritom je E energijska funkcija definirana izrazom (94), dok $Z(\boldsymbol{X})$ predstavlja normalizacijsku konstantu.

$$E(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\substack{i \\ i \in \{1,2,\dots,n\}}} \psi_u(y_i|\boldsymbol{x}) + \sum_{\substack{i<j \\ i,j \in \{1,2,\dots,n\}}} \psi_b(y_i, y_j|\boldsymbol{x}) \tag{94}$$

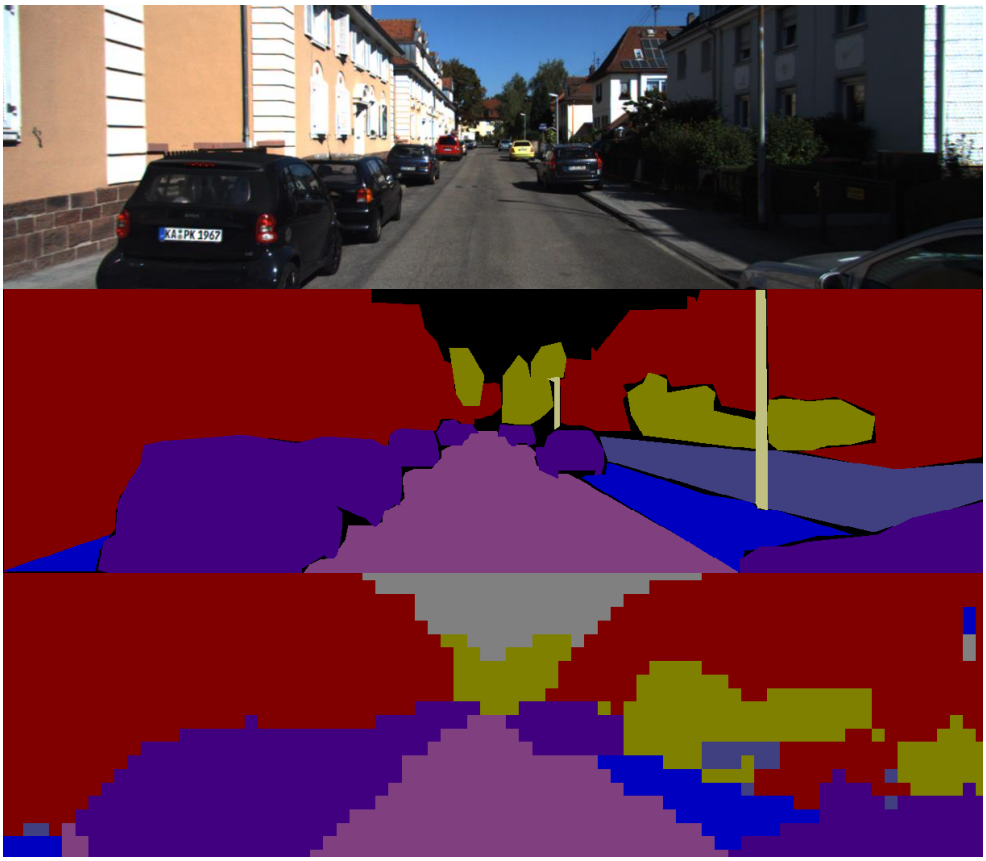
Izraz (94) predstavlja energiju, tj. cijenu preslikavanja ulazne slike $\boldsymbol{x}$ u semantičku mapu $\boldsymbol{y}$. Unarni potencijali označeni su s ψ_u, a binarni s ψ_b. Unarni i binarni potencijali mogu poprimiti pozitivne i negativne vrijednosti, te predstavljaju cijenu klasifikacije piksela i, odnosno para piksela (i, j) u semantičke razrede y_i odnosno (y_i, y_j). Unarni potencijali definirani su nezavisno za svaki piksel, te modeliraju nezavisnu cijenu razreda s obzirom na lokalno susjedstvo promatranog piksela. U implementiranom sustavu, unarni potencijali definirani su kao negativni logaritam izlaza kontekstnog uvjetnog slučajnog polja. Ako se izlaz kontekstnog uvjetnog slučajnog polja nakon bilinearne interpolacije za piksel i označi sa $Q_i(Y_i|\boldsymbol{x})$, onda je pripadni unarni potencijal definiran kao:

$$\psi_u(Y_i|\boldsymbol{x}) = -\ln Q_i(Y_i|\boldsymbol{x}) \tag{95}$$

Na ovaj način je osigurano da ψ_u predstavlja cijenu dodjeljivanja oznake semantičkog razreda pikselu. Ako je Q_i poprimao maksimalnu vrijednost za $Y_i = y_i$, onda će pripadna vrijednost $\psi_u(y_i)$ biti manja od svih ostalih vrijednosti $\psi_u(y_j), j \neq i$.

5.2.1.1. Binarni potencijali

Budući da unarni potencijali opisuju klasifikaciju piksela u semantičke razrede neovisno jedan o drugom, njihova predikcija je općenito nekonzistentna i šumovita (slika 16).



Slika 16 Nekonzistentnost predikcije temeljene isključivo na unarnim potencijalima. Odozgo prema dolje prikazane su: izvorna slika, ručno označena slika, izlaz dobiven primjenom unarnih potencijala

Binarni potencijali rješavaju navedeni problem modeliranjem cijene supojavljivanja bliskih piksela s obzirom na njihovu poziciju i boju. Spomenuti potencijali mogu se modelirati na sljedeći način [8]:

$$\psi_b(y_i, y_j|\boldsymbol{X}) = \mu(y_i, y_j) \sum_{m=1}^{K} w^{(m)} k^{(m)}(\boldsymbol{f}_i, \boldsymbol{f}_j) = \mu(y_i, y_j) k(\boldsymbol{f}_i, \boldsymbol{f}_j) \tag{96}$$

$\mu(y_i, y_j)$ je funkcija prikladnosti oznaka semantičkih razreda. Definirana je na sljedeći način:

$$\mu(y_i, y_j) = \begin{cases} 1, & ako\ y_i \neq y_j \\ 0, & inače \end{cases} \tag{97}$$

Funkcija prikladnosti definira kaznu za susjedne piksele koji su klasificirani u različite semantičke razrede.

$k^{(m)}$ je Gaussova jezgra (98) definirana nad vektorima značajki f_i i f_j koji opisuju piksele i i j. $w^{(m)}$ je težina pridijeljena jezgri $k^{(m)}$ prilikom njihove linearne kombinacije.

$$k^{(m)}(f_i, f_j) = e^{-\frac{1}{2}(\boldsymbol{f}_i - \boldsymbol{f}_j)^T \Sigma_{(m)}^{-1} (\boldsymbol{f}_i - \boldsymbol{f}_j)} \tag{98}$$

$\Sigma_{(m)}$ je pozitivno definitna matrica kovarijanci koja opisuje oblik jezgre $k^{(m)}$.

Implementirani sustav koristi dvije vrste Gaussovih jezgri za definiranje binarnih potencijala:

- jezgra zaglađivanja
- jezgra izgleda

Jezgra zaglađivanja definirana je nad vektorima značajki $\boldsymbol{f}_i, \boldsymbol{f}_j$ koji uključuju pozicije piksela i i j. Matrica kovarijanci definirana je kao dijagonalna matrica. Pritom varijanca θ_γ^2 definira širinu susjedstva te utječe na konačnu vrijednost kazne.

$$\begin{aligned} k^{(1)}(\boldsymbol{f}_i, \boldsymbol{f}_j) = k^{(1)}(\boldsymbol{p}_i, \boldsymbol{p}_j) &= e^{-\frac{1}{2}\left(\begin{bmatrix} p_{xi} \\ p_{yi} \end{bmatrix} - \begin{bmatrix} p_{xj} \\ p_{yj} \end{bmatrix}\right)^T \begin{bmatrix} \theta_\gamma^2 & 0 \\ 0 & \theta_\gamma^2 \end{bmatrix}^{-1} \left(\begin{bmatrix} p_{xi} \\ p_{yi} \end{bmatrix} - \begin{bmatrix} p_{xj} \\ p_{yj} \end{bmatrix}\right)} \\ &= e^{-\frac{\|\boldsymbol{p}_i - \boldsymbol{p}_j\|^2}{2\theta_\gamma^2}} \end{aligned} \tag{99}$$

Jezgra zaglađivanja modelira izglednost da bliski pikseli pripadaju istom semantičkom razredu. Pritom parametar θ_γ modelira stupanj bliskosti piksela.

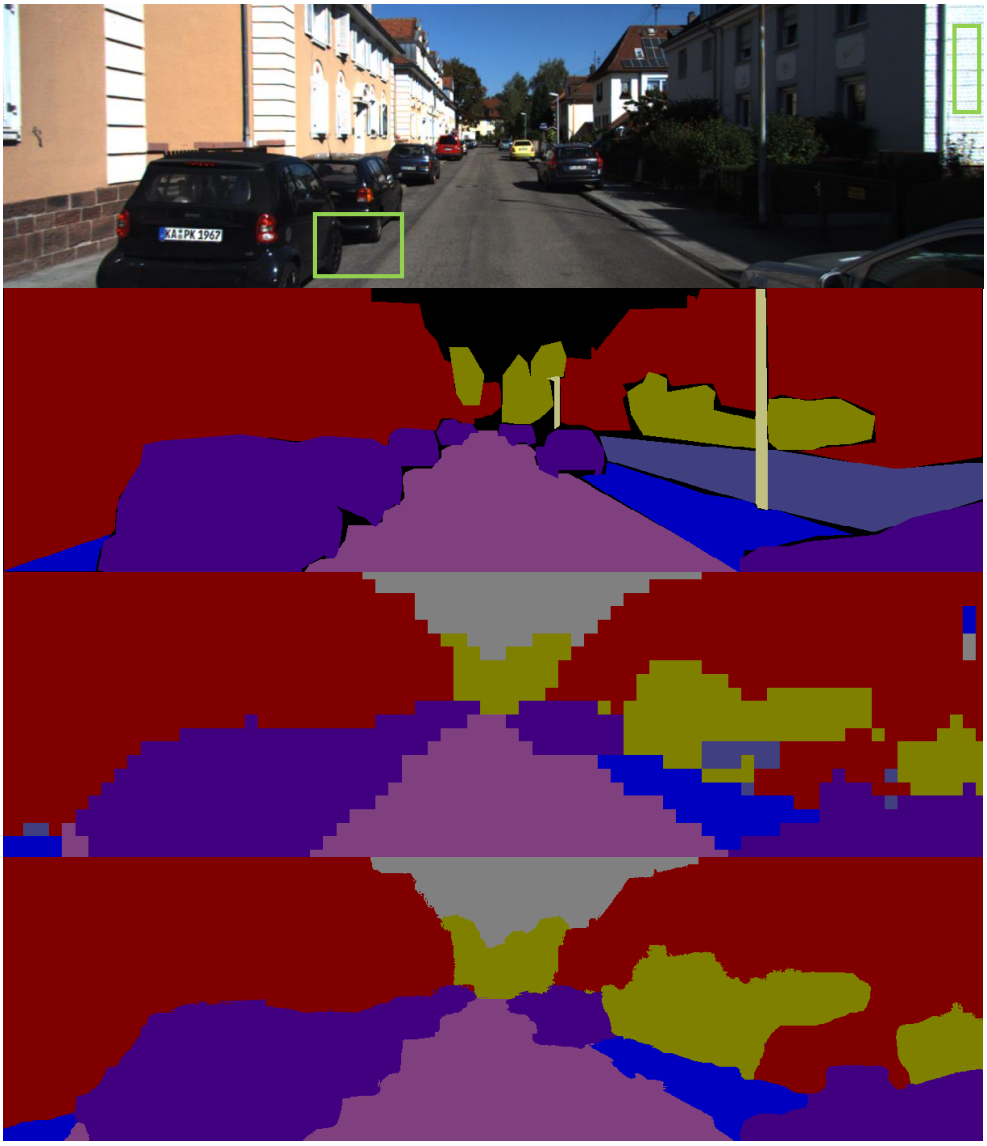
Za razliku od jezgre zaglađivanja, jezgra izgleda definirana je nad vektorima značajki koji uz pozicije piksela uključuju i njihove boje:

$$\boldsymbol{f}_i = \begin{bmatrix} p_{xi} \\ p_{yi} \\ x_{ir} \\ x_{ig} \\ x_{ib} \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_i \\ \boldsymbol{x}_i \end{bmatrix} \tag{100}$$

Matrica kovarijanci definirana je kao dijagonalna matrica, te uključuje varijance θ_α^2 i θ_β^2, koje respektivno opisuju širine susjedstava u domeni pozicija odnosno boja piksela. Izraz (101) prikazuje definiciju jezgre izgleda.

$$\begin{aligned} k^{(2)}(\boldsymbol{f}_i, \boldsymbol{f}_j) = e &\quad -\frac{1}{2}\left(\begin{bmatrix} \boldsymbol{p}_i \\ \boldsymbol{x}_i \end{bmatrix} - \begin{bmatrix} \boldsymbol{p}_j \\ \boldsymbol{x}_j \end{bmatrix}\right)^T \begin{bmatrix} \theta_\alpha^2 & 0 & 0 & 0 & 0 \\ 0 & \theta_\alpha^2 & 0 & 0 & 0 \\ 0 & 0 & \theta_\beta^2 & 0 & 0 \\ 0 & 0 & 0 & \theta_\beta^2 & 0 \\ 0 & 0 & 0 & 0 & \theta_\beta^2 \end{bmatrix}^{-1} \left(\begin{bmatrix} \boldsymbol{p}_i \\ \boldsymbol{x}_i \end{bmatrix} - \begin{bmatrix} \boldsymbol{p}_j \\ \boldsymbol{x}_j \end{bmatrix}\right) \\ &= e^{-\frac{\|\boldsymbol{p}_i - \boldsymbol{p}_j\|^2}{2\theta_\alpha^2} - \frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\theta_\beta^2}} \end{aligned} \tag{101}$$

Jezgra izgleda modelira izglednost da bliski pikseli slične boje pripadaju istom semantičkom razredu. Stupanj bliskosti kontrolira parametar θ_α, a stupanj sličnosti boja parametar θ_β. Jezgra zaglađivanja pomaže u uklanjanju izoliranih regija, a jezgra izgleda u boljem definiranju bridova u semantičkoj mapi (slika 17).



Slika 17 Utjecaj binarnih potencijala. Odozgo prema dolje prikazane su: izvorna slika, ručno označena slika, unarni potencijali dobiveni kontekstnim uvjetnim slučajnim poljem, te izlaz dobiven primjenom unarnih i binarnih potencijala.

Binarnim potencijalima modelira se sljedeća situacija. Neka su dva susjedna piksela klasificirana u različite semantičke razrede. Što su navedeni pikseli bliži jedan drugome, jezgra zaglađivanja pridjeljuje veću kaznu takvoj klasifikaciji. Analogno, što su pikseli bliži i sličniji u boji, jezgra izgleda također pridjeljuje veću kaznu takvoj klasifikaciji. Na taj način se potiče drugačija klasifikacija spomenutih piksela koja će smanjiti ukupnu vrijednost energijske funkcije (94).

5.2.2. Učenje

Učenje potpuno povezanog uvjetnog slučajnog polja definiranog nad pikselima svodi se na učenje parametara $w^{(1)}$, θ_γ, $w^{(2)}$, θ_α i θ_β binarnih potencijala budući da su unarni potencijali već naučeni tijekom obrade slika na gruboj rezoluciji (učenje generatora potencijala i primjena kontekstnog uvjetnog slučajnog polja). Spomenuti parametri uče se pretragom po rešetci. Po uzoru na [9], najprije se $w^{(1)}$ i θ_γ fiksiraju, te se vrši gruba pretraga po rešetci za preostala tri parametra. Nakon toga primjenjuje se finija pretraga po rešetci za svih pet parametara. Detalji provedbe pretrage po rešetci opisani su u poglavlju 6.

5.2.3. Zaključivanje

Kako u gusto povezanom uvjetnom slučajnom polju definiranom nad pikselima $\boldsymbol{Y}$ čini potpuno povezani graf, tj. svaki piksel je povezan sa svim ostalim pikselima u slici, efikasna implementacija zaključivanja zahtijeva primjenu metode srednjeg polja. Budući da je distribucija spomenutog uvjetnog slučajnog polja definirana preko energijske funkcije i potencijala ψ_u i ψ_b, izraz (93) je moguće prilagoditi općenitoj distribuciji (27) analogno kao u poglavlju 5.1.5:

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \frac{1}{Z(\boldsymbol{X})} e^{-\left[\sum_i \psi_u(Y_i|\boldsymbol{X}) + \sum_{i<j} \psi_b(Y_i, Y_j|\boldsymbol{X})\right]} = \frac{1}{Z(\boldsymbol{X})} \prod_i e^{-\psi_u(Y_i|\boldsymbol{X})} \prod_{i<j} e^{-\psi_b(Y_i, Y_j|\boldsymbol{X})} \tag{102}$$

Pritom je $i, j \in \{1,2,\ldots,n\}$. Vidljivo je kako faktori ϕ iz (27) odgovaraju eksponentima negativnih potencijala ψ_u i ψ_b. Iterativna jednadžba metode srednjeg polja za gusto povezano uvjetno slučajno polje definirano nad pikselima definira se na sljedeći način:

$$
\begin{aligned}
Q_i(y_i|\boldsymbol{x}) &= \frac{1}{Z_i(\boldsymbol{x})}\exp\left\{\sum_{\phi:Y_i\in\boldsymbol{Y}_\phi} E_{(\boldsymbol{Y}_\phi-\{Y_i\})\sim Q}\left[\ln\phi\left(\boldsymbol{Y}_\phi, y_i|\boldsymbol{x}\right)\right]\right\} \\
&= \frac{1}{Z_i(\boldsymbol{x})}\exp\left\{E_{(\{Y_i\}-\{Y_i\})\sim Q}[-\psi_u(y_i|\boldsymbol{x}_i)]\right. \\
&\left. + \sum_{j\neq i} E_{Y_j\sim Q}\left[-\psi_b(y_i, Y_j|\boldsymbol{x})\right]\right\} \\
&= \frac{1}{Z_i(\boldsymbol{x})}\exp\left\{-\psi_u(y_i|x_i) - \sum_{j\neq i}\sum_{y_j\in Val(Y_j)} Q_j(y_j|\boldsymbol{x})\psi_b(y_i, y_j|\boldsymbol{x})\right\}
\end{aligned}
\tag{103}
$$

U izrazu (103) pretpostavlja se upotreba svojstva simetričnosti binarnih potencijala $\psi_b(y_i, y_j|\boldsymbol{x}) = \psi_b(y_j, y_i|\boldsymbol{x})$ u zbroju $\sum_{j\neq i} E_{Y_j\sim Q}[-\psi_b(y_i, Y_j|\boldsymbol{x})]$ tako da se zbroje vrijednosti svih binarnih potencijala definiranih između čvora i i svih njegovih susjeda. Uzevši u obzir definiciju binarnih potencijala (96) dobiva se konačna iterativna jednadžba srednjeg polja:

$$
\begin{aligned}
Q_i(y_i|\boldsymbol{x}) = \frac{1}{Z_i(\boldsymbol{x})}\exp\Bigg\{&-\psi_u(y_i|x_i) \\
&- \sum_{y_j\in Val(Y_j)} \mu\left(y_i, y_j\right)\sum_{m=1}^{K} w^{(m)}\sum_{j\neq i} k^{(m)}(\boldsymbol{f}_i, \boldsymbol{f}_j)\, Q_j(y_j|\boldsymbol{x})\Bigg\}
\end{aligned}
\tag{104}
$$

Pseudokod prilagođenog algoritma srednjeg polja prikazan je u tablici 8.

Tablica 8 Prilagođeni algoritam srednjeg polja za potpuno povezano uvjetno slučajno polje definirano nad pikselima

Postavke:

- $i, j \in \{1,2,\dots,n\}$
- $m \in \{1,2\}$
- $y_i, l \in Val(Y_i)$

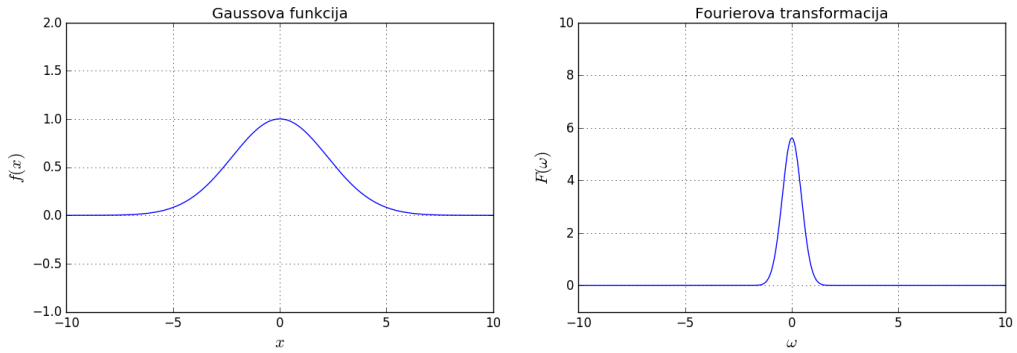
1. Inicijalizacija: $Q_i(y_i|\boldsymbol{x}) \leftarrow \exp\{-\psi_u(y_i|\boldsymbol{x}_i)\}$ za $\forall i, \forall y_i$
   a. *Normaliziraj* $Q_i(Y_i|\boldsymbol{x})$ *za* $\forall i$
2. Ponavljaj zadani broj iteracija:
   a. $\tilde{Q}_i^{(m)}(l|\boldsymbol{x}) \leftarrow \sum_{j\neq i} k^{(m)}(\boldsymbol{f}_i, \boldsymbol{f}_j)\, Q_j(l|\boldsymbol{x})$ za $\forall i, m, l$
   b. $\hat{Q}_i(y_i|\boldsymbol{x}) \leftarrow \sum_{l\in Val(Y_j)} \mu(y_i, l)\sum_m w^{(m)}\, \tilde{Q}_i^{(m)}(l|\boldsymbol{x})$ za $\forall i, y_i$
   c. $Q_i(y_i|\boldsymbol{x}) \leftarrow \exp\{-\psi_u(y_i|\boldsymbol{x}) - \hat{Q}_i(y_i|\boldsymbol{x})\}$ za $\forall i, y_i$
   d. *Normaliziraj* $Q_i(Y_i|\boldsymbol{x})$ *za* $\forall i$

Svaka iteracija prethodno opisanog algoritma odvija se kroz četiri koraka. Računski najzahtjevniji je korak razmjene poruka (korak 2a). Razmjena poruka omogućuje trenutnom pikselu (čvoru u grafu) prikupljanje informacija o semantičkim oznakama svih njegovih susjeda. Naivna implementacija zahtijevala bi kvadratno vrijeme izvođenja ($O(n^2)$). Primjena funkcije prikladnosti (korak 2b) i lokalno ažuriranje (korak 2c) mogu se izvesti u linearnom vremenu. Budući da je u uvjetnom slučajnom polju definiranom nad pikselima svaki piksel povezan sa svim ostalim pikselima, razmjena poruka se mora efikasnije izvesti. Specifična definicija binarnih potencijala preko Gaussovih funkcija omogućava efikasnu izvedbu razmjene poruka pomoću konvolucije.

Korak razmjene poruka može se prikazati kao:

$$\tilde{Q}_i^{(m)}(l|\boldsymbol{x}) = \sum_{j \neq i} k^{(m)}(\boldsymbol{f}_i, \boldsymbol{f}_j)\, Q_j(l|\boldsymbol{x}) = \sum_{j} k^{(m)}(\boldsymbol{f}_i, \boldsymbol{f}_j) Q_j(l|\boldsymbol{x}) - Q_i(l|\boldsymbol{x}) \\ = \left[G_{\Sigma_{(\mathrm{m})}} * Q(l|\boldsymbol{x})\right](\boldsymbol{f}_i) - Q_i(l|\boldsymbol{x}) = \bar{Q}_i^{(m)}(l|\boldsymbol{x}) - Q_i(l|\boldsymbol{x}) \tag{105}$$

Pritom $\bar{Q}_i^{(m)}(l|\boldsymbol{x})$ predstavlja konvoluciju Gaussove jezgre $G_{\Sigma_{(\mathrm{m})}}$ s mapom vjerojatnosti $Q(l|\boldsymbol{x})$. Gaussova jezgra je dvodimenzionalna mapa koja se dobije izračunom vrijednosti pripadne jezgre $k^{(m)}$ između piksela i i svih njegovih susjeda. Konvolucija sa Gaussovom jezgrom efektivno predstavlja nisko-propusni filtar koji će odbaciti sve visoke frekvencije u $Q(l|\boldsymbol{x})$, te na taj način pojasno ograničiti $\bar{Q}_i^{(m)}(l|\boldsymbol{x})$. Slika 18 prikazuje jednodimenzionalnu Gaussovu funkciju u prostornoj i frekvencijskoj domeni. Može se primijetiti kako širi Gauss (veća standardna devijacija) u prostornoj domeni rezultira užim Gaussom u frekvencijskoj domeni, što dovodi do odsijecanja visokih frekvencija prilikom konvolucije s drugim signalom u prostornoj domeni. Analogno razmatranje vrijedi i za dvodimenzionalnu Gaussovu funkciju.



Slika 18 Gaussova funkcija i njezina Fourierova transformacija [14]

Uzevši u obzir Shannonov teorem uzorkovanja [13][11], koji kaže kako se izvorni signal može savršeno rekonstruirati iz uzorkovanog signala ukoliko je frekvencija uzorkovanja veća ili jednaka od dvostruke maksimalne frekvencije uzorkovanog signala, odabirom prikladne frekvencije uzorkovanja moguće je rekonstruirati $\bar{Q}_i^{(m)}(l|\boldsymbol{x})$ na temelju njegovih uzoraka. Paris i Durand [10] su eksperimentalno pokazali kako se konvolucija (105) može izvesti pod-uzorkovanjem $Q(l|\boldsymbol{x})$ uz razmak između uzoraka proporcionalan standardnoj devijaciji Gaussove jezgre $G_{\Sigma_{(\mathrm{m})}}$. Nad dobivenim uzorcima moguće je primijeniti konvoluciju, te dobiveni rezultat natrag nad-uzorkovati. Opisani postupak prikazuje tablica 9.

Tablica 9 Efikasna izvedba razmjene poruka

1. $Q_{\downarrow}(l\|\boldsymbol{x}) \leftarrow pod\_uzorkuj(Q(l\|\boldsymbol{x}))$
2. $\bar{Q}_{\downarrow i}^{(m)}(l\|\boldsymbol{x}) \leftarrow \sum_{\downarrow j} k^{(m)}(\boldsymbol{f}_{\downarrow i}, \boldsymbol{f}_{\downarrow j}) Q_{\downarrow j}(l\|\boldsymbol{x})$ za $\forall \downarrow i$
3. $\bar{Q}^{(m)}(l\|\boldsymbol{x}) \leftarrow nad\_uzorkuj\left(\bar{Q}_{\downarrow}{}^{(m)}(l\|\boldsymbol{x})\right)$

Kako vrijednost Gaussove funkcije veoma brzo opada s povećanjem udaljenosti, koristi se aproksimacija Gaussove jezgre čije su vrijednosti koje odgovaraju udaljenostima većim od dvije standardne devijacije postavljene na nulu. Aproksimacija Gaussove jezgre smanjuje vremensku složenost izvedbe konvolucije budući da omogućava njezinu aproksimaciju agregiranjem konstantnog broja susjednih vrijednosti.

5.2.3.1. Permutoedarska rešetka

Za efikasnu implementaciju razmjene poruka (tablica 9) koristi se permutoedarska rešetka (engl. permutohedral lattice). Permutoedarska rešetka je efikasna konvolucijska podatkovna struktura koja omogućava izvedbu visoko

dimenzionalnog filtriranja u vremenu $O(nd^2)$. Pritom je s d označena dimenzionalnost vektora značajki $\boldsymbol{f}_i$, dok n predstavlja broj piksela u slici. Permutoedarska rešetka koristi se za izvedbu konvolucije [12]:

$$\bar{Q}_i(l|\boldsymbol{x}) = \sum_j e^{-\frac{1}{2}\left\|\tilde{\boldsymbol{f}}_i - \tilde{\boldsymbol{f}}_j\right\|^2} Q_j(l|\boldsymbol{x}) \tag{106}$$

Pritom je kovarijacijska matrica Gaussove konvolucijske jezgre jedinična dijagonalna matrica, što se jednostavno postiže transformacijom prostora značajki na sljedeći način:

$$\tilde{\boldsymbol{f}} = U\boldsymbol{f} \tag{107}$$

Matrica U dobiva se dekompozicijom kovarijacijske matrice Gaussove jezgre:

$$\Sigma_{(m)} = UU^T \tag{108}$$

Permutoedarska rešetka koristi svojstvo separabilnosti Gaussovih jezgri kako bi visoko dimenzionalnu konvoluciju rastavila na niz jednodimenzionalnih konvolucija uzduž osi rešetke.

d-dimenzionalna permutoedarska rešetka definira se kao projekcija skalirane $(d+1)$ dimenzionalne cjelobrojne rešetke $(d+1)\mathbb{Z}^{d+1}$ duž vektora $\mathbf{1} = [1,1,\dots,1]^T$ na hiperravninu $H_d: \boldsymbol{x} \cdot \mathbf{1} = 0$. Suma koordinata svih točaka u H_d je jednaka 0. Permutoedarska rešetka dijeli prostor H_d uniformnim simpleksima pri čemu je za svaku točku u H_d moguće odrediti unutar kojeg simpleksa se nalazi. Vrhovi d-dimenzionalnog simpleksa definirani su kao:

$$s_k = [\overbrace{k,\dots,k}^{d+1-k}, \overbrace{k-(d+1),\dots,k-(d+1)}^{k}] \tag{109}$$

Projekcije vektora baze rešetke $(d+1)\mathbb{Z}^{d+1}$ definirane su kao $[-1,\dots,1,d,1,\dots,-1]^T$, te one opisuju minimalne pomake između točaka u prostoru H_d.

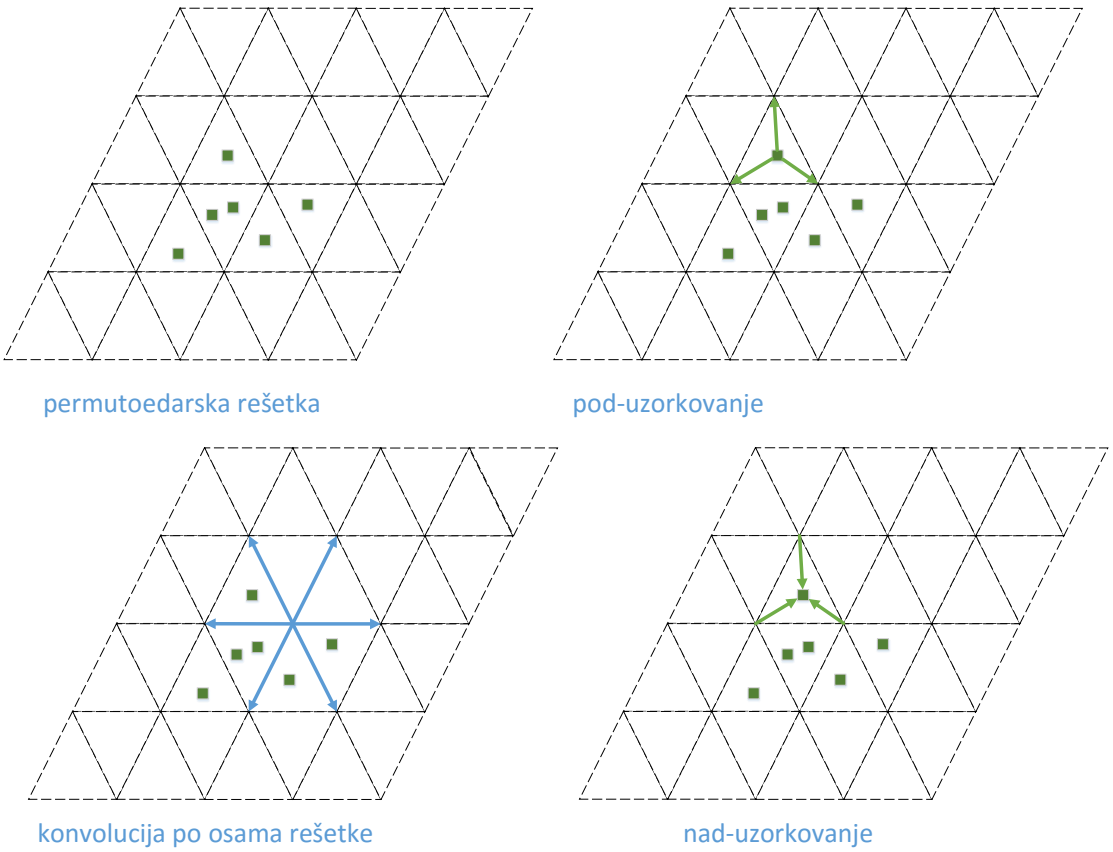
Konvolucija pomoću permutoedarske rešetke izvodi se u nekoliko koraka:

1. Preslikavanje vektora značajki $\tilde{\boldsymbol{f}}_i$ na H_d primjenom ortogonalne baze B od H_d.
2. Pod-uzorkovanje (*engl. splatting*) – za svaki vektor značajki u H_d, odrede se njegove baricentrične koordinate (110), te se $b_k Q_i(l|\boldsymbol{x})$ spremi u pripadni vrh simpleksa koji okružuje preslikani vektor značajki $\boldsymbol{t}_i = B\tilde{\boldsymbol{f}}_i$.

$$b_j = \frac{\boldsymbol{t}_{i,d-j} - \boldsymbol{t}_{i,d-j+1}}{d+1}, b_o = 1 - \sum_{j=1}^{d} b_j \qquad (110)$$

3. Konvolucija (*engl. blurring*) – Uzduž svake dimenzije permutoedarske rešetke određene vektorima $\pm[1,..,1,-d,1,...,1]^T$ primijeni se konvolucija sa jednodimenzionalnom Gaussovom jezgrom.
4. Nad-uzorkovanje (*engl. slicing*) – Primjenom baricentričnih težina obavlja se obrnuti postupak od pod-uzorkovanja. Na izvornim pozicijama vektora značajki $\tilde{\boldsymbol{f}}_i$ primjenom baricentričnih koordinata rekonstruira se $\bar{Q}_i(l|\boldsymbol{x})$ na temelju vrijednosti, koje su prohranjene u vrhovima okružujućeg simpleksa.

Prethodno opisani koraci 2-4 odgovaraju koracima 1-3 iz tablice 9. Koraci izvedbe razmjene poruka pomoću permutoedarske rešetke prikazani su na slici 19. U gornjem lijevom kutu prikazana je permutoedarska rešetka s uzorcima u višedimenzionalnom prostoru. U gornjem desnom kutu prikazan je korak pod-uzorkovanja, dok je nad-uzorkovanje prikazano u donjem desnom kutu. U donjem lijevom kutu prikazane su konvolucije duž osi permutoedarske rešetke.



Slika 19 Koraci postupka filtriranja pomoću permutoedarske matrice [15]

6. Eksperimentalni rezultati

6.1. Skupovi slika

Za potrebe evaluacije implementiranog postupka semantičke segmentacije korištena su dva javno dostupna skupa slika: KITTI i Cityscapes.

6.1.1. KITTI

Skup podataka KITTI [25] sadrži veliku kolekciju slika i video isječaka snimljenih u Karlsruheu i njegovoj bližoj okolici. Izvorni skup podataka prvenstveno je namijenjen rješavanju problema stereo podudaranja, optičkog toka, vizualne odometrije, te praćenja i detekcije 3D objekata. Skup podataka izrađen je u sklopu zajedničkog projekta dva instituta: *Karlsruhe Institute of Technology* i *Toyota Technological Institute*. Za semantičku segmentaciju prilagođen je maleni dio izvornog skupa slika. German Ros [25] je ručno označio 146 slike. Navedeni skup proširen je s 299 dodatnih ručno označenih slika [26]. Korišteni skup sadrži ukupno 445 slika podijeljenih u tri skupa: skup za učenje (353), skup za validaciju (46) i skup za ispitivanje (46 slika). Skup za ispitivanje sastoji se od istih slika koje Ros definira kao skup za ispitivanje, dok je skup za validaciju generiran nasumičnim izborom 46 od preostalih 399 slika. Sve slike su dimenzija $1241x376$ piksela. Skup podataka sadrži 11 semantičkih razreda čija je zastupljenost prikazana u tablici 10.

Tablica 10 Udio semantičkih razreda u skupu podataka KITTI

[%]	skup za učenje	skup za validaciju	skup za ispitivanje
nebo	6.85	8.41	3.63
zgrada	20.96	19.00	32.84
cesta	17.74	19.17	13.71
pločnik	7.69	5.77	7.33
ograda	2.87	3.52	6.11
vegetacija	36.49	35.84	18.80
stup	0.53	0.60	0.41
auto	6.31	7.13	16.64
prometni znak	0.36	0.48	0.11
pješak	0.06	0.05	0.22
biciklist	0.14	0.03	0.20

Primjeri slika iz opisanog skupa prikazane su u nastavku (slika 20).

Slika 20 KITTI - primjeri izvornih(lijevo) i ručno označenih slika(desno)

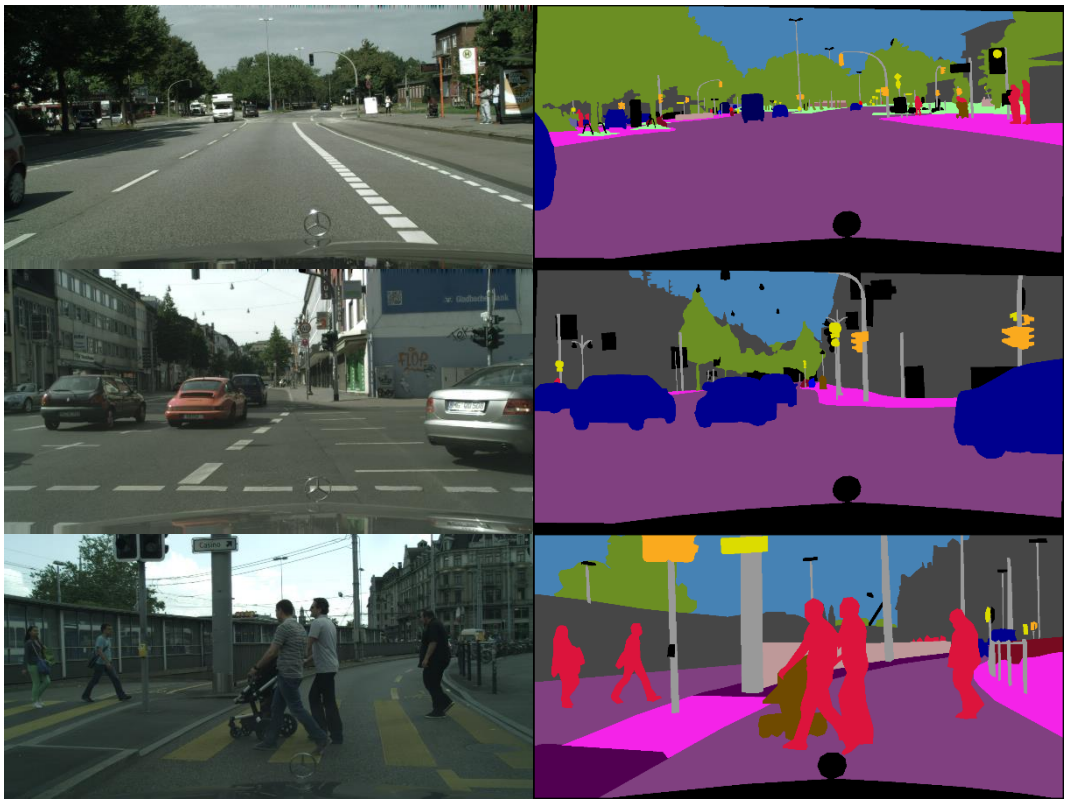
6.1.2. Cityscapes

Skup podataka Cityscapes [27] izrađen je s fokusom na semantičko razumijevanje urbanih scena. Slike su snimljene u 50 njemačkih gradova tijekom više mjeseci (u proljeće, ljeto i jesen). Sve slike snimljene su tijekom dana u dobrim ili srednje dobrim vremenskim uvjetima. Skup podataka sadrži 5000 detaljno označenih, te 20000 grubo označenih slika. U sklopu ovog rada korištene su samo detaljno označene slike. Slike su dimenzija $2048x1024$ piksela. Podijeljene su u tri skupa: skup za učenje (2975), skup za validaciju (500) i skup za ispitivanje (1525 slika). Budući da oznake za skup za ispitivanje nisu javno dostupne, u sklopu ovog rada skup za validaciju (500 slika) korišten je kao skup za ispitivanje, dok je skup za učenje podijeljen na skup za validaciju (487) i učenje (preostalih 2488 slika). Podjela na skup za validaciju i učenje izvršena je nasumičnim izborom gradova, čime se dobro aproksimira skup za ispitivanje. Skup podataka sadrži 19 semantičkih razreda čija je zastupljenost prikazana u tablici 11.

Tablica 11 Udio semantičkih razreda u skupu podataka Cityscapes

[%]	skup za učenje	skup za validaciju	skup za ispitivanje
cesta	36.53	38.6	37.65
pločnik	6.40	4.43	5.40
zgrada	23.41	19.72	21.92
zid	0.71	0.39	0.73
ograda	0.90	0.77	0.82
stup	1.16	1.62	1.49
semafor	0.19	0.31	0.20
prometni znak	0.52	0.69	0.66
vegetacija	16.08	15.12	17.32
teren	1.16	1.15	0.83
nebo	3.90	4.65	3.36
osoba	1.00	2.38	1.30
motociklist	0.12	0.24	0.22
auto	6.84	7.81	6.51
kamion	0.22	0.50	0.30
bus	0.15	0.66	0.39
vlak	0.25	0.16	0.11
motor	0.10	0.11	0.08
bicikl	0.36	0.69	0.71

U nastavku su prikazani primjeri slika iz opisanog skupa (slika 21).



Slika 21 Cityscapes - primjeri izvornih(lijevo) i ručno označenih slika(desno)

6.2. Metrike

Za potrebe evaluacije semantičke segmentacije mogu se koristiti različite metrike. U nastavku se opisuje konfuzijska matrica, na temelju koje su definirane metrike korištene u ovom radu.

Konfuzijska matrica je tablica čiji retci predstavljaju predviđene instance, a stupci stvarne instance objekata. Pojam objekata u kontekstu semantičke segmentacije odnosi se na piksele pojedinog semantičkog razreda. Na temelju konfuzijske matrice mogu se očitati četiri karakteristične vrijednosti, koje se koriste za bolju procjenu performansi naučenog modela:

- TP_i (*engl. true positives*) – broj piksela koji pripadaju razredu i, te su klasificirani u razred i.
- TN_i (*engl. true negatives*) – broj piksela koji ne pripadaju razredu i, te su klasificirani u neki razred $j \neq i$.
- FP_i (*engl. false positives*) – broj piksela koji ne pripadaju razredu i, ali su klasificirani u razred i.
- FN_i (*engl. false negatives*) – broj piksela koji pripadaju razredu i, ali su klasificirani u neki razred $j \neq i$.

Primjer konfuzijske matrice za slučaj tri semantička razreda prikazan je u tablici 12.

Tablica 12 Primjer konfuzijske matrice za tri semantička razreda

$stvarno \rightarrow$ / $predviđeno \downarrow$	**mačka**	**pas**	**miš**
mačka	384	22	11
pas	12	300	0
miš	0	0	100

Ako se promatra semantički razred „*mačka*“, mogu se iz konfuzijske matrice pročitati sljedeće vrijednosti:

- $TP_{mačka} = 384$
- $TN_{mačka} = 300 + 0 + 0 + 100 = 400$
- $FP_{mačka} = 22 + 11 = 33$
- $FN_{mačka} = 12 + 0 = 12$

Na temelju prethodno opisanih vrijednosti mogu se definirati sljedeće metrike:

- Točnost piksela (*engl. pixel accuracy*) – udio ispravno klasificiranih piksela u ukupnom broju piksela (111). j označava proizvoljno odabrani razred, a K ukupan broj semantičkih razreda.

$$Acc_{pixel} = \frac{\sum_{i=1}^{K} TP_i}{TP_j + FP_j + FN_j + TN_j} \tag{111}$$

- Odziv (*engl. recall*) razreda – udio ispravno klasificiranih piksela u skupu svih piksela razreda i.

$$recall_i = \frac{TP_i}{TP_i + FN_i} \tag{112}$$

- Preciznost (*engl. precision*) razreda – udio ispravno klasificiranih piksela u skupu piksela klasificiranih u razred i.

$$precision_i = \frac{TP_i}{TP_i + FP_i} \tag{113}$$

- IoU (*engl. intersection over union*) razreda – udio ispravno klasificiranih piksela razreda i u skupu svih piksela koji su klasificirani kao razred i ili stvarno pripadaju razredu i.

$$IoU_i = \frac{TP_i}{TP_i + FP_i + FN_i} \tag{114}$$

Preciznost, odziv i IoU mogu se usrednjiti po razredima, čime se dobiva jedna brojčana vrijednost koja opisuje performansu naučenog modela.

$$\begin{aligned} recall_{mean} &= \frac{1}{K}\sum_{i=1}^{K} recall_i \\ precision_{mean} &= \frac{1}{K}\sum_{i=1}^{K} precision_i \\ IoU_{mean} &= \frac{1}{K}\sum_{i=1}^{K} IoU_i \end{aligned} \tag{115}$$

6.3. Eksperimentalni rezultati

Svi eksperimenti provođeni su na skupu podataka Cityscapes. Konačna performansa implementiranog modela evaluirana je na oba skupa podataka: KITTI i Cityscapes.

Implementirani model učen je u dva koraka:

1. Učenje unarnih i binarnih potencijala
2. Učenje parametara uvjetnog slučajnog polja definiranog nad pikselima

Za učenje parametara generatora potencijala koristi se Adam inačica stohastičkog gradijentnog spusta. Početna stopa učenja postavljena je na 10^{-4}, te se eksponencijalno smanjuje svake 3 epohe s faktorom 0.5. Izbor početne stope učenja proveden je unakrsnom validacijom. Svi modeli uče se 15 epoha, te se odabire onaj model, koji ostvaruje najveći $mIoU$ na validacijskom skupu (487 slika). Kako bi se izbjegla prenaučenost modela, koristi se regularizacijski faktor iznosa 0.005. Implementirani model (tablica 5 i tablica 6) koristi grupnu normalizaciju. Međutim, isključeno je učenje parametara β i γ te se vrši samo obična normalizacija. Za potrebe izračuna funkcije gubitka koriste se ručno označene slike koje su smanjene na rezoluciju reducirane slike (izlaza generatora značajki) metodom najbližih susjeda.

Nakon učenja unarnih i binarnih potencijala, pomoću kontekstnog uvjetnog slučajnog polja generirane su mape vjerojatnosti koje se nad-uzorkuju (bilinearnom interpolacijom) do izvorne rezolucije, te koriste kao unarni potencijali uvjetnog slučajnog polja definiranog nad pikselima. Koristi se pretraga po rešetci kako bi se odredili parametri spomenutog uvjetnog slučajnog polja definiranog nad pikselima. Pritom se najprije obavlja gruba pretraga: $(w^{(1)}, \theta_\gamma)$ su fiksirani na $(3{,}3)$, a ostali parametri pretražuju u sljedećim rasponima:

- $w^{(2)} \in \{2{,}3{,}5{,}10\}$
- $\theta_\alpha \in \{50{,}60{,}70{,}80{,}90{,}100\}$
- $\theta_\beta \in \{3{,}4{,}5{,}6{,}7{,}8{,}9{,}10\}$

Prilikom pretrage, različite konfiguracije parametara evaluiraju se na podskupu validacijskog skupa (100 slika za Cityscapes, čitav validacijski skup za KITTI) zbog vremenske složenosti pretrage. Nakon završetka grube pretrage vrši se finija pretraga (pretražuje se uže susjedstvo dotad pronađene konfiguracije parametara).

6.3.1. Eksperimenti s kontekstnim uvjetnim slučajnim poljem

Svi eksperimenti opisani u nastavku provedeni su na slikama smanjene rezolucije ($608x304$) iz skupa Cityscapes. Naredni eksperimenti opisuju performanse kontekstnog uvjetnog slučajnog polja. Svi brojevi dobiveni su evaluacijom na slikama reducirane rezolucije (16 puta smanjena rezolucija). Pritom je broj iteracija metode srednjeg polja kod kontekstnog uvjetnog slučajnog polja ograničen na 3.

Tablica 13 prikazuje performanse modela sa i bez grupne normalizacije. Vidljivo je kako grupna normalizacija omogućava bržu i bolju konvergenciju modela. Broj epoha bio je ograničen na 15. Prikazani su rezultati za slučaj uporabe unarnih i binarnih potencijala za susjedstvo „okruženje“ uz veličinu susjedstva $3x3$.

Tablica 13 Doprinos grupne normalizacije

model – unarni + okruženje $3x3$ [%]	$\boldsymbol{IoU_{mean}}$
bez grupne normalizacije	53.40
s grupnom normalizacijom	$\mathbf{54.40}$

Zbog malene rezolucije reducirane slike, koja se koristi prilikom izračuna funkcije gubitka, kao i velikog kapaciteta modela, model se može vrlo jednostavno

pretrenirati. Opisani problem riješen je primjenom regularizacijskog faktora. Isprobana je i dropout metoda (tablica 14). Dropout je primijenjen na slojeve $conv6\_1$, $conv6\_2$, $conv6\_3$, $convu\_1$ i $convb\_1$. Korištena je relacija susjedstva „okruženje“ veličine $3x3$. Pokazuje se kako dropout slabije pomaže od L_2 regularizacije, te je u konačnom modelu korištena isključivo L_2 regularizacija.

Tablica 14 Performanse na skupovima za učenje i validaciju - dropout i L_2 regularizacija

model – unarni + okruženje $3x3$ [%]	**IoU_{mean} – skup za validaciju**	**IoU_{mean} – skup za učenje**
samo L_2 regularizacija	**54.40**	**83.90**
dropout ($conv6\_1$, $conv6\_2$)	53.93	87.27
dropout ($conv6\_1$, $conv6\_2$, $conv6\_3$, $convu\_1$, $convb\_1$)	52.20	87.57

Prilikom učenja mreže u svim prethodnim eksperimentima primijenjena je funkcija gubitka (89). Navedena funkcija fokusirana je na poboljšanje točnosti piksela, a ne usrednjenog presjeka nad unijom (IoU_{mean}). Prilikom učenja, slabije zastupljenim razredima se ne pripisuje veći značaj, te je modelu sa stajališta točnosti piksela isplativije naučiti dobro klasificirati velike razrede, uslijed čega IoU_{mean} opada. Spomenuti efekt posebno je izražen kod učenja oba tipa binarnih potencijala budući da unarni i binarni potencijali dijele početni dio mreže. U takvoj konfiguraciji mreži postaje dosta teško postići jednaku razinu presjeka nad unijom kao kod primjene isključivo unarnih potencijala. Tablica 15 opisuje prethodno spomenute efekte učenja bez balansiranja razreda. Konfiguracija eksperimenata opisana je u tablici. Vidljiv je pad usrednjenog presjeka nad unijom kod unarnih potencijala kada se model uči zajedno s binarnim potencijalima. Kod modela s binarnim potencijalima također je prikazana performansa kada se prilikom zaključivanja koristi samo jedna vrsta potencijala.

Tablica 15 Utjecaj učenja modela bez balansiranja razreda

konfiguracija	veličina susjedstva	potencijali kod zaključivanja	IoU_{mean}
unarni	-		$\mathbf{53,14\%}$
unarni + okruženje	$5x5$	unarni	$\mathbf{52,57\%}$
		svi	$53,70\%$
unarni + iznad/ispod	$5x5$	unarni	$\mathbf{52,85\%}$
		svi	$54,05\%$
unarni + okruženje + iznad/ispod	$5x5, 5x5$	unarni	$\mathbf{51,84\%}$
		okruženje	$53,42\%$
		Iznad/ispod	$53,03\%$
		svi	$53,30\%$

S ciljem rješavanja prethodno opisanog problema, primijenjena je normalizirana funkcija gubitka s balansiranjem razreda (116). Navedena funkcija povećava značaj manje zastupljenih razreda prilikom učenja, te na taj način ostvaruje bolje performanse sa stajališta usrednjenog presjeka nad unijom (IoU_{mean}).

$$\frac{\lambda}{2}\|\theta\|_2^2 - \sum_{i=1}^{|D|}\left[\frac{1}{|N_U|}\sum_{p\in N_U}\frac{1}{P_{pr}^{(i)}(y_p)}\ln P_U(y_p|\boldsymbol{x}^i,\boldsymbol{\theta}_U) + \frac{1}{N_o}\sum_{p\in N_U, q\in S_o(p)}\frac{1}{P_{pr}^{(i)}(y_p)P_{pr}^{(i)}(y_q)}\ln P_{V_o}(y_p,y_q|\boldsymbol{x}^i,\boldsymbol{\theta}_{V_o}) + \frac{1}{N_i}\sum_{p\in N_U, q\in S_i(p)}\frac{1}{P_{pr}^{(i)}(y_p)P_{pr}^{(i)}(y_q)}\ln P_{V_i}(y_p,y_q|\boldsymbol{x}^i,\boldsymbol{\theta}_{V_i})\right] \quad (116)$$

N_i i N_o predstavljaju ukupan broj parova susjednih piksela za relaciju „iznad/ispod“ odnosno „okruženje“. $P_{pr}^{(i)}(y_p)$ predstavlja apriornu vjerojatnost razreda y_p. Spomenuta vjerojatnost definira se kao omjer broja piksela razreda y_p i ukupnog broja piksela u i-toj slici. Apriorna vjerojatnost para razreda (y_p, y_q) aproksimirana je naivnom pretpostavkom kao produkt apriornih vjerojatnosti pojedinačnih razreda. Svi modeli u nastavku učeni su primjenom funkcije gubitka (116).

Tablica 16 prikazuje utjecaj veličine i vrste susjedstva na konačnu performansu modela. Vidljivo je kako veće susjedstvo rezultira poboljšanom performansom modela. Najveće poboljšanje postiže se primjenom oba tipa binarnih potencijala uz veličinu susjedstva $7x7$.

Tablica 16 Utjecaj veličine susjedstva na performanse modela

konfiguracija	veličina susjedstva	potencijali kod zaključivanja	IoU_{mean}
unarni	-	unarni	53,34%
unarni + okruženje	$3x3$	unarni	53,51%
		svi	54,69%
	$5x5$	unarni	53,13%
		svi	54,91%
	$7x7$	unarni	53,41%
		svi	54,92%
unarni + iznad/ispod	$3x3$	unarni	53,56%
		svi	54,76%
	$5x5$	unarni	53,77%
		svi	55,00%
	$7x7$	unarni	53,56%
		svi	55,09%
unarni + okruženje + iznad/ispod	$3x3, 3x3$	unarni	53,47%
		okruženje	55,16%
		Iznad/ispod	54,78%
		svi	55,03%
	$5x5, 5x5$	unarni	53,41%
		okruženje	55,21%
		Iznad/ispod	55,26%
		svi	55,27%
	$7x7, 7x7$	unarni	53,86%
		okruženje	55,49%
		Iznad/ispod	55,52%
		svi	**55,57%**

Budući da su ulazne slike dimenzija $608x304$, reducirani izlaz generatora značajki bit će dimenzija $38x19$. Navedena rezolucija otežava učenje modela budući da se model zbog velikog kapaciteta lako može pretrenirati. Također, doprinosi različitih veličina susjedstava nisu toliko izraženi kao što bi to bio slučaj s većom rezolucijom. Učenje modela provedeno je uz 25% veće dimenzije ulaznih slika ($768x384$) kako bi se vidio utjecaj na performanse. Prilikom učenja korišteno je susjedstvo veličine $7x7$ tipa „iznad/ispod" kako bi se eksperiment mogao provesti relativno brzo. Poboljšanje performansi ostvareno povećanjem rezolucije prikazano je u tablici 17.

Tablica 17 Utjecaj rezolucije izvorne slike na performanse modela

model – unarni + iznad/ispod $7x7$ [%]	$\boldsymbol{IoU_{mean}}$
$\mathbf{608x304}$	55.09
$\mathbf{768x384}$	$\mathbf{57.63}$

Skup podataka za učenje moguće je dodatno proširiti zrcaljenjem slika, čime se otežava postizanje prenaučenosti, te se istovremeno podiže performansa modela budući da se model uči na mnogo većem skupu podataka. Tablica 18 prikazuje doprinos zrcaljenja slika prilikom učenja modela. Eksperiment je proveden za susjedstvo „okruženje" veličine $3x3$ i slike rezolucije $608x304$. Vidljivo je kako se postiže značajno poboljšanje performansi.

Tablica 18 Utjecaj proširenja skupa za učenje zrcaljenim slikama

model – unarni + okruženje $3x3$ [%]	$\boldsymbol{IoU_{mean}}$ **– skup za validaciju**	$\boldsymbol{IoU_{mean}}$ **– skup za učenje**
bez zrcaljenja	54.40	83.90
s zrcaljenjem	$\mathbf{56.14}$	83.36

6.3.2. Rezultati

Konačni rezultati dobiveni su primjenom oba tipa susjedstava veličine $7x7$. Prilikom učenja korišteno je balansiranje razreda, grupna normalizacija kao i zrcaljenje slika. Model je treniran na slikama rezolucije $768x384$ za skup slika Cityscapes, te $608x192$ za skup KITTI. Broj iteracija srednjeg polja kod kontekstnog uvjetnog slučajnog polja ograničen je na 3, a kod uvjetnog slučajnog polja definiranog nad

pikselima na 10. Nakon učenja generatora potencijala, generirane su vjerojatnosne mape pomoću kontekstnog uvjetnog slučajnog polja. Spomenute mape bilinearnom su interpolacijom nad-uzorkovane do izvorne rezolucije, te korištene za definiranje unarnih potencijala uvjetnog slučajnog polja definiranog nad pikselima. Dobiveni rezultati evaluirani su na slikama izvorne rezolucije. Uslijed nad-uzorkovanja, dolazi do pada performansi sustava. Kako bi se spomenuti efekt ublažio, prilikom učenja generatora potencijala odabran je onaj model koji ostvaruje najbolji IoU na validacijskim slikama izvorne rezolucije.

6.3.2.1. Cityscapes

Rezultati koje implementirani sustav ostvaruje na skupu za ispitivanje prikazani su u tablici 19. U tablici je najprije prikazana performansa kontekstnog uvjetnog slučajnog polja, a potom i konačna performansa sustava (nakon primjene uvjetnog slučajnog polja definiranog nad pikselima). Bitno je naglasiti kako su performanse kontekstnog uvjetnog slučajnog polja izračunate na slikama izvorne rezolucije zbog čega su dobivene brojke različite od onih iz tablice 17.

Tablica 19 Rezultati na skupu za ispitivanje - Cityscapes

[%]	**potencijali kod zaključivanja**	IoU_{mean}	***točnost piksela***	***odziv***	***preciznost***
kontekstno uvjetno slučajno polje	unarni	50.76	88.40	67.32	62.95
	okruženje	51.55	88.59	65.18	66.76
	iznad/ispod	51.61	88.59	65.34	66.96
	svi	51.74	88.61	65.38	67.06
uvjetno slučajno polje definirano nad pikselima	unarni + Gaussovi binarni	**54.52**	90.36	66.23	71.62

Tablica 20 prikazuje IoU ostvaren na skupu za ispitivanje za svaki semantički razred. Prikazan je IoU, kojeg ostvaruju kontekstno, te uvjetno slučajno polje

definirano nad pikselima. Također je prikazan i IoU kojeg ostvaruju unarni potencijali kontekstnog uvjetnog slučajnog polja. Vidljivo je kako uvjetno slučajno polje s Gaussovim binarnim potencijalima definiranim nad pikselima značajno doprinosi poboljšanju performansi.

Tablica 20 IoU po semantičkim razredima - Cityscapes

IoU [%]	unarni potencijali	kontekstno uvjetno slučajno polje	uvjetno slučajno polje definirano nad pikselima
cesta	95.10	94.48	95.21
pločnik	68.08	64.88	68.92
zgrada	78.48	79.91	82.84
zid	33.25	37.19	39.42
ograda	32.69	33.58	35.48
stup	16.66	18.10	19.85
semafor	29.83	27.52	28.90
prometni znak	36.24	37.90	44.50
vegetacija	81.57	80.85	84.13
teren	47.94	48.46	52.60
nebo	81.98	79.76	86.35
osoba	52.33	51.86	56.08
motociklist	32.62	31.31	32.39
auto	81.59	79.47	82.18
kamion	34.00	37.90	39.23
bus	53.44	59.41	61.35
vlak	26.22	36.56	38.05
motor	32.10	32.75	35.18
bicikl	50.43	51.13	53.26

Primjeri segmentiranih slika prikazani su u nastavku (slika 22). Odozgo prema dolje prikazane su: izvorna slika, ručno označena slika, izlaz uz unarne potencijale, izlaz uz unarne potencijale i jezgru zaglađivanja, izlaz uz unarne potencijale i jezgru izgleda, izlaz konačnog modela. Na slikama je vidljiv doprinos jezgri zaglađivanja i

izgleda uklanjanju izoliranih regija, te poboljšanju segmentacije uz rubove objekata. Pojedini objekti, poput stupova izgledaju preširoko, te premaleni objekti nisu uspješno detektirani. Navedeni problem može se riješiti promatranjem izvornih slika pri različitim rezolucijama [28][26]. Segmentacijska performansa dodatno je smanjena zbog učenja mreže na smanjenoj rezoluciji. Povećanjem rezolucije očekivano je i značajno povećanje performansi sustava.

Slika 22 Cityscapes - segmentirane slike. Odozgo prema dolje prikazane su: izvorna slika, ručno označena slika, izlaz uz unarne potencijale, izlaz uz unarne potencijale i jezgru zaglađivanja, izlaz uz unarne potencijale i jezgru izgleda, izlaz konačnog modela.

6.3.2.2. KITTI

Performanse implementiranog sustava prikazane su u tablici 21. U tablici je najprije prikazana performansa kontekstnog uvjetnog slučajnog polja, a potom i konačna performansa sustava (nakon primjene uvjetnog slučajnog polja definiranog nad pikselima).

Tablica 21 Rezultati na skupu za ispitivanje - KITTI

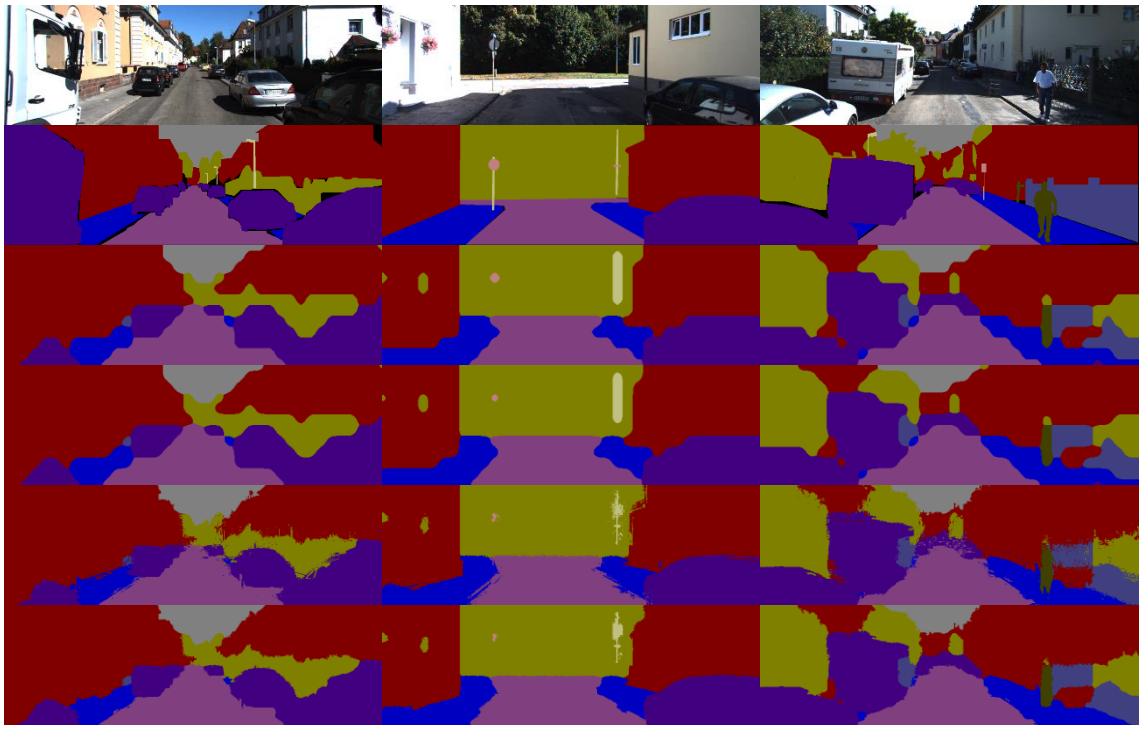
[%]	**potencijali kod zaključivanja**	IoU_{mean}	***točnost piksela***	***odziv***	***preciznost***
kontekstno uvjetno slučajno polje	unarni	45.28	79.45	60.22	58.88
	okruženje	50.37	84.60	57.48	72.33
	iznad/ispod	50.48	84.61	57.61	72.63
	svi	50.53	84.62	57.66	72.84
uvjetno slučajno polje definirano nad pikselima	unarni + Gaussovi binarni	**52.34**	86.11	58.43	78.27

Tablica 22 prikazuje IoU ostvaren na skupu za ispitivanje za svaki semantički razred. Prikazan je IoU, kojeg ostvaruju kontekstno, te uvjetno slučajno polje definirano nad pikselima. Također je prikazan i IoU kojeg ostvaruju unarni potencijali kontekstnog uvjetnog slučajnog polja. Vidljivo je kako uvjetno slučajno polje s Gaussovim binarnim potencijalima definiranim nad pikselima poboljšava ukupnu performansu sustava.

Tablica 22 IoU po semantičkim razredima - KITTI

IoU [%]	unarni potencijali	kontekstno uvjetno slučajno polje	uvjetno slučajno polje definirano nad pikselima
nebo	67.86	70.92	79.73
zgrada	75.69	78.39	80.38
cesta	74.86	88.00	89.20
pločnik	53.36	68.42	70.32
ograda	38.23	37.37	38.27
vegetacija	67.31	73.12	76.29
stup	6.40	10.85	11.05
auto	64.39	72.06	73.62
prometni znak	14.98	21.57	22.08
pješak	24.21	23.73	23.18
biciklist	10.78	11.43	11.61

Primjeri segmentiranih slika prikazani su u nastavku (slika 23). Odozgo prema dolje prikazane su: izvorna slika, ručno označena slika, izlaz uz unarne potencijale, izlaz uz unarne potencijale i jezgru zaglađivanja, izlaz uz unarne potencijale i jezgru izgleda, izlaz konačnog modela. Vidljiv je doprinos jezgre zaglađivanja smanjenju izoliranih područja, kao i doprinos jezgre izgleda boljoj segmentaciji rubova objekata.



Slika 23 KITTI - segmentirane slike. Odozgo prema dolje prikazane su: izvorna slika, ručno označena slika, izlaz uz unarne potencijale, izlaz uz unarne potencijale i jezgru zaglađivanja, izlaz uz unarne potencijale i jezgru izgleda, izlaz konačnog modela.

6.4. Implementacijski detalji

Generator potencijala i kontekstno uvjetno slučajno polje implementirani su u programskom jeziku python, te radnom okviru Tensorflow [29]. Tensorflow je otvorena biblioteka za numeričke proračune, temeljena na grafovima toka podataka. Izvođenje složene operacije svodi se na izgradnju grafa čiji čvorovi predstavljaju jednostavnije operacije, a bridovi višedimenzionalne tenzore koje pojedine operacije generiraju, a druge primaju kao ulaze. Fleksibilna arhitektura Tensorflowa omogućava jednostavno definiranje mjesta izvođenja pojedinih operacija (CPU ili GPU). Tensorflow su izvorno razvili inženjeri i istraživači iz *Google Brain* tima za provedbu istraživanja u području strojnog učenja i dubokih neuronskih mreža. Iako se Tensorflow većinom koristi za duboko učenje (tako i u ovom radu), dovoljno je općenit za primjenu i u drugim domenama. Za razliku od generatora potencijala, uvjetno slučajno polje definirano nad pikselima implementirano je u programskom jeziku C++ [30].

U sklopu rada napisan je još niz pomoćnih skripti za generiranje i prikaz rezultata opisanih u ovom radu. Budući da je sav kod javno dostupan[1], te veoma dobro dokumentiran, neće biti posebno opisana njegova primjena u sklopu samog rada.

[1] https://github.com/Vaan5/piecewisecrf

7. Zaključak

Semantička segmentacija predstavlja veoma zanimljivo i zahtjevno područje računalnog vida u kojem su grafički modeli, a prvenstveno uvjetna slučajna polja našla široku primjenu.

Pokazano je kako konvolucijske neuronske mreže predstavljaju moćan radni okvir za detekciju značajki u slikama, te se uspješno mogu iskoristiti i za učenje binarnih potencijala uvjetnog slučajnog polja. Uvjetno slučajno polje u stanju je relativno efikasno kombinirati unarne i binarne potencijale i za veća susjedstva primjenom metode srednjeg polja. Prednost primijenjenog postupka učenja konvolucijskih mreža, koje definiraju binarne potencijale leži u izostanku potrebe za zaključivanjem tijekom učenja. Zaključivanje, iako razmjerno brzo zbog metode srednjeg polja, značajno bi usporilo postupak učenja gdje je potrebno obaviti ogroman broj iteracija stohastičkog gradijentnog spusta. U radu je eksperimentalno pokazano kako metoda balansiranja razreda kao i normalizacija, pomažu prilikom učenja binarnih potencijala, te ublažuju negativni efekt slabo zastupljenih razreda u skupu podataka. Gusto povezano uvjetno slučajno polje s Gaussovim binarnim potencijalima definiranim nad bojom i pozicijama piksela uspješno je iskorišteno kao postprocesirajući postupak za uklanjanje šumovite segmentacije na rubovima objekata, te uklanjanje izoliranih regija iz segmentirane slike. Implementirani model evaluiran je na javno dostupnim skupovima podataka za semantičku segmentaciju: KITTI i Cityscapes.

Tensorflow se pokazao kao veoma dobar i fleksibilan radni okvir za duboko učenje. Budući da se radi o otvorenom projektu, njegova buduća proširenja (poput dilatiranih konvolucijskih slojeva i boljeg upravljanja memorijom) omogućit će dodatna poboljšanja implementiranog sustava.

Buduća nadogradnja implementiranog sustava uključivala bi uporabu više skala ulazne slike [28] ili pak primjenu invarijantne reprezentacije ulazne slike [26] čime bi se značajno podigla performansa modela. Također bi vrijedilo eksperimentirati s drugim vrstama susjedstava, primjerice „lijevo/desno“, kojima bi se modelirale dodatne relacije između semantičkih oznaka razreda. Kako bi se sustav mogao primijeniti i na slike veoma velike rezolucije, potrebno je implementaciju metode

srednjeg polja optimizirati čime bi se dodatno ubrzalo zaključivanje kod kontekstnog uvjetnog slučajnog polja.

Literatura

[1] Koller, D., Friedman N. Probabilistic graphical models: Principles and techniques. MIT Press, 2009.

[2] Elezović, N. Diskretna vjerojatnost. 5. izdanje. Zagreb: Element, 2012.

[3] Šnajder, J., Bašić, B.D. Strojno učenje. Zagreb, 2013.

[4] German, S., German, D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 6(1984), str. 721-741.

[5] Lafferty, J., McCallum, A., Pereira, F.C.N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proceedings of the 18th International Conference on Machine Learning, Williamstown, (2001), str. 282-289.

[6] Lin, G., Shen, C., Reid, I. Efficient piecewise training of deep structured models for semantic segmentation. IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, (2016).

[7] Simonyan, K., Zisserman, A. Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations, San Diego, (2015).

[8] Krähenbühl, P., Koltun, V. Efficient inference in fully connected crfs with Gaussian edge potentials. Neural Information Processing Systems, Granada, (2011), str. 109-117.

[9] Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. International Conference on Learning Representations, San Diego, (2015).

[10] Paris, S., Durand, F. A fast approximation of the bilateral filter using a signal processing approach. International Journal of Computer Vision. 81, 1(2009), str. 24-52.

[11] Smith, S.W. The scientist and engineer's guide to digital signal processing. San Diego: California Technical Publishing, 1997.

[12] Adams, A. Baek, J., Davis, M.A. Fast high-dimensional filtering using the permutohedral lattice. Computer Graphics Forum. 29, 2(2010), str. 753-762.

[13] Shannon, C.E. Communication in the presence of noise. Proceedings of the Institute of Radio Engineers. 37, 1(1949), str. 10-21.

[14] Jeren, B. Predavanja s predmeta Signali i Sustavi, 27.3.2013., https://www.fer.unizg.hr/predmet/sis2, pristupljeno 10.6.2016.

[15] Koltun, V. Dense crf – slides, 14.12.2011., Efficient inference in fully connected crfs with Gaussian edge potentials, http://vladlen.info/papers/densecrf-slides.pdf, pristupljeno 10.6.2016.

[16] Badrinarayanan, V., Kendall, A., Cipolla, R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint, arXiv:1511.00561, 2015.

[17] McCulloch, W.S., Pitts, W. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 5, (1943), str. 127-147.

[18] Čupić, M., Šnajder, J., Bašić, B.D. Umjetne neuronske mreže. Zagreb, 2008.

[19] Rosenblatt, F. The perceptron: A perceiving and recognizing automaton. New York: Cornell Aeronautical Laboratory, 1957.

[20] Hubel, D., Wiesel, T. Receptive fields and functional architecture of monkey striate cortex. Journal of Physiology (London), 195, (1968), str. 215–243.

[21] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salkhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. Journal on Machine Learning Research, 15, 1(2014), str. 1929-1958.

[22] Ioffe, S., Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd International Conference on Machine Learning, Lille, (2015), str. 448-456.

[23] Kingma, D.P., Ba, J.L. Adam: A method for stochastic optimization. arXiv preprint, arXiv:1412.6980, 2014.

[24] Čupić, M., Bašić, B.D., Golub, M. Neizrazito, evolucijsko I neuroračunarstvo. Zagreb, 2013.

[25] Ros, G., Ramos, S., Granados, M., Bakhtiary, A., Vazquez, D., Lopez, A.M. Vision-based offline-online perception paradigm for autonomous driving. IEEE Winter Conference on Applications of Computer Vision, Hawaii, (2015), str. 231-238.

[26] Krešo, I., Čaušević, D., Krapac, J., Šegvić, S. Convolutional scale invariance for semantic segmentation. 38th German Conference on Pattern Recognition, Hannover, (2016).

[27] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, SS., Schiele, B. The Cityscapes dataset for semantic urban scene understanding. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, (2016).

[28] Farabet, C., Couprie, C., Najman, L., LeCun, Y. Learning hierarchical features for scene labeling. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35, 8(2013), str. 1915-1929.

[29] Abadi, M. i ostali. Tensorflow: Large-scale machine learning on heterogeneous systems, 2015, https://www.tensorflow.org/, pristupljeno 10.6.2016.

[30] Krähenbühl, P., Koltun, V. Dense crf – code, Efficient inference in fully connected crfs with Gaussian edge potentials, http://googledrive.com/host/0B6qziMs8hVGieFg0UzE0WmZaOW8/code/densecrf.zip, pristupljeno 10.6.2016.

[31] Yu, H.J, More about undirected graphical models, 2010., *Probabilistic models – lectures*, https://www.cs.helsinki.fi/group/cosco/Teaching/Probability/2010/lecture5_MRF2.pdf, pristupljeno 27.6.2016.

Modeliranje supojavljivanja semantičkih oznaka uvjetnim slučajnim poljima

Sažetak

Uvjetna slučajna polja su diskriminativni probabilistički grafički model strojnog učenja s mnogim primjenama u klasifikaciji strukturiranih podataka. U ovom radu primjenjuju se za rješavanje problema semantičke segmentacije. Ostvarivanje zadovoljavajućih performansi kod semantičke segmentacije zahtijeva modeliranje supojavljivanja semantičkih oznaka. Implementirani sustav modelira spomenute relacije pomoću dva uvjetna slučajna polja. Prvo polje koristi binarne potencijale naučene od strane konvolucijske neuronske mreže za modeliranje dvije vrste susjedstava – „okruženje“ i „iznad/ispod“. Drugo polje je gusto povezano, te koristi Gaussove binarne potencijale definirane nad vektorom boja i pozicijama piksela, kako bi poboljšalo segmentacijski rezultat prethodnog uvjetnog slučajnog polja. Implementirani model evaluiran je na standardnim skupovima u području razumijevanja urbanih prometnih scena (Cityscapes i KITTI).

Ključne riječi: konvolucijske neuronske mreže, uvjetna slučajna polja, metoda srednjeg polja, Gaussovi binarni potencijali

Modelling the Co-Occurrence of Semantic Labels with Conditional Random Fields

Abstract

Conditional random fields are probabilistic discriminative graphical machine learning models with a wide variety of applications for structured prediction. The focus of this paper is their usage for semantic segmentation. Achieving satisfiable semantic segmentation performance requires semantic label co-occurrence modelling. The implemented system models such relations with two conditional random fields. The first one uses binary potentials learned by a convolutional neural network for modelling two types of relations – “surrounding” and “above/below”. The second field is a dense conditional random field with Gaussian binary potentials defined on pixel positions and color vectors. It is used for post-processing the segmentation result of the previous random field. The implemented model is evaluated on publicly available datasets for urban traffic scene understanding (Cityscapes and KITTI).

Keywords: convolutional neural networks, conditional random fields, mean field approximation, Gaussian binary potentials