SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2288

PREDVIĐANJE VIŠEMODALNE SEMANTIČKE BUDUĆNOSTI U VIDEU

Kristijan Fugošić

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2288

PREDVIĐANJE VIŠEMODALNE SEMANTIČKE BUDUĆNOSTI U VIDEU

Kristijan Fugošić

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 13. ožujka 2020.

DIPLOMSKI ZADATAK br. 2288

Pristupnik:	**Kristijan Fugošić (0036494267)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	prof. dr. sc. Siniša Šegvić

Zadatak: **Predviđanje višemodalne semantičke budućnosti u videu**

Opis zadatka:

Predviđanje semantičke budućnosti u videu neriješen je problem računalnog vida s mnogim zanimljivim primjenama. U posljednje vrijeme najbolji rezultati u tom području postižu se dubokim konvolucijskim modelima. Ovaj rad razmatra rješenje tog problema uz pomoć generativnih modela koji su u stanju modelirati višemodalnu budućnost uvjetovanu slikama iz prošlosti scene. U okviru rada, potrebno je proučiti konvolucijske arhitekture za semantičko predviđanje cestovnih scena u videu. Oblikovati generativni model za predviđanje višemodalne budućnosti u videu. Predložiti metriku za mjerenje uspješnosti modela. Validirati hiperparametre, prikazati i ocijeniti ostvarene rezultate te provesti usporedbu s rezultatima iz literature, sve na skupu Cityscapes. Predložiti pravce budućeg razvoja. Radu priložiti izvorni kod razvijenih postupaka uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 30. lipnja 2020.

Zahvaljujem se svojoj obitelji na velikoj podršci tijekom cijelog školovanja, asistentu Josipu Šariću na pomoći pri izradi rada i mentoru prof. dr. sc. Siniši Šegviću na svoj pomoći i korisnim savjetima u protekle tri godine.

SADRŽAJ

1. Uvod

Samovozeći automobili goruća su tema današnjice. Svojim dolaskom promijenit će način na koji gledamo na putnički i teretni promet. U potpunosti će izmijeniti javni prijevoz i taxi usluge, a s vremenom će dovesti u pitanje isplativost posjedovanje vlastitog automobila. No, kako bismo riješili jedan tako kompleksan zadatak, prvo moramo riješiti niz jednostavnijih. Jedan od bitnijih elemenata sustava za autonomnu vožnju jest prepoznavanje okoline i njeno razumijevanje. Veoma je bitno da sustav zna prepoznati cestu, pješake koji se kreću uz kolnik ili po njemu, automobil ispred sebe i sve ostale sudionike u prometu, što semantičku segmentaciju čini vrlo popularnim problemom.

Mogućnost predviđanja budućnosti također je važan atribut inteligentnog ponašanja. Intuitivno je jasno da bi sustavi poput onih za autonomnu vožnju profitirali gledanjem u prošlost i predviđanjem neposredne budućnosti prilikom donošenja odluka, umjesto promatranja isključivo trenutnog stanja. Čak i modeli koji to uzimaju u obzir, problemu pristupaju konzervativno, dodjeljujući najviše prostora automobilima, cesti, nebu i sličnim razredima koji često zauzimaju velik dio slike i za koje je lakše predvidjeti gdje će se nalaziti u skorijoj budućnosti. Takvim pristupom prostor se nerijetko oduzima manjim objektima kao što su znakovi, stupovi i pješaci, čije je kretanje veoma bitno predvidjeti. Položaj manjih objekata, ponajviše udova pješaka, teško je predvidjeti, stoga je ideja ovog rada dopustiti modelu nešto više slobode u smislu mogućnosti predviđanja više različitih budućnosti. Dodatnu motivaciju za to pronalazimo i u situacijama gdje se otkriva novi prostor, primjerice kod skretanja ili prolaskom drugog vozila. Ponekad pogledom u blisku prošlost možemo zaključiti što se u novootkrivenom prostoru nalazi, a nekada jednostavno ne možemo znati. U oba slučaja htjeli bismo da naš model pokuša pogoditi, uvažavajući sve moguće buduće scenarije. To ćemo pokušati postići pretvaranjem osnovnog regresijskog modela u uvjetni generativni model temeljen na suparničkom učenju uz korištenje gubitaka rekonstrukcije momenata.

Ovaj rad je koncipiran tako da u ranijim poglavljima objašnjavamo pojmove, tehnike i modele koje ćemo kasnije koristiti. Pa tako odmah nakon uvoda, u drugom

poglavlju, započinjemo s definiranjem pojmova semantičke segmentacije i segmentacije instanci. Kroz treće poglavlje u kratkim crtama su opisane konvolucije i konvolucijski slojevi, te nešto složeniji deformirajući konvolucijski slojevi, a u nastavku je teorijski obrađen put od modela R-CNN do modela Mask R-CNN. U četvrtom poglavlju opisana je osnovna ideja rezidualnih neuronskih mreža, dok su u petom i šestom predstavljeni osnovni principi generativnih i uvjetnih suparničkih modela. Kroz sedmo poglavlje analiziramo ideju koja predlaže upotrebu drugačijih rekonstrukcijskih gubitaka s ciljem sprječavanja kolapsa modova kod generativnih suparničkih modela. U osmom poglavlju teorijski obrađujemo model za predviđanje budućnosti na razini značajki, dok u devetom poglavlju kombiniramo različite modele, tehnike i prijedloge iz prijašnjih poglavlja kako bismo dizajnirali vlastiti model, čije rezultate prikazujemo u zadnjem, desetom, poglavlju uz prethodan opis korištenih metrika.

2. Semantička segmentacija i segmentacija instanci

Jedan od najpopularnijih problema današnjice na kojemu se zadnjih godina užurbano radi je autonomna vožnja, a semantička segmentacija je bitan korak koji automobilima pomaže pri raspoznavanju okoline i ispravnom kretanju. Semantičku segmentaciju možemo opisati kao postupak raspoznavanja različitih objekata na slici na razini piksela. Cilj semantičke segmentacije je svrstati svaki piksel na slici u jedan od fiksnog broja razreda. Budući da se ovaj rad fokusira na scene iz prometa, na slici 2.1 prikazan je primjer iz skupa Cityscapes[1].

Slika 2.1: Primjer označene slike iz skupa za učenje Cityscapes. Različiti objekti su označeni različitim bojama, pa tako na ovom konkretnom primjeru možemo razlikovati zgrade, kolnik, pločnik, automobile, ljude, tramvaj, drveća, stupove, semafore i znakove. Na slici su vidljiva i dječja kolica, koja ne spadaju u osnovne razrede već se vode kao ostali dinamični objekti, te zastave i rasvjeta koji spadaju pod ostale statične objekte.

Još zanimljiviji i nešto teži problem je segmentacija instanci. Dok će semantička segmentacija svrstati sve objekte istog razreda (npr. sve ljude na slici) u istu kategoriju, segmentacija instanci prepoznaje pojedini objekt (svaki čovjek je označen drugačije).

(a) Primjer semantičke segmentacije - svi ljudi su pridijeljeni istom razredu

(b) Primjer segmentacije instanci - svaki čovjek je zaseban razred

Slika 2.2: Razlika između semantičke segmentacije i segmentacije instanci, kako je prikazano u [17].

Osim u prometu, semantička segmentacija i segmentacija instanci koriste se i u robotskim i sigurnosnim sustavima, virtualnoj i proširenoj stvarnosti te sličnim sustavima koji imaju potrebu raspoznati objekte. Velikom uspjehu na ovom području pridonijele su metode dubokog učenja, naročito konvolucijske neuronske mreže koje su opisane u sljedećem poglavlju.

3. Konvolucijske neuronske mreže

Konvolucijske neuronske mreže su podvrsta neuronskih mreža. Zahtijevaju minimalnu ili nikakvu prethodnu obradu slika te imaju sposobnost samostalno naučiti značajke koje su se u tradicionalnim algoritmima ručno osmišljale. Zbog sposobnosti iskorištavanja hijerarhije u podacima te slaganja složenijih uzoraka iz jednostavnijih često su korištene na području računalnog vida. U nastavku ćemo prvo opisati osnovni gradivni element konvolucijskih mreža - konvolucijski sloj, te njegovu nešto složeniju inačicu - deformirajući konvolucijski sloj. Nakon toga ćemo obraditi modele za detekciju i segmentaciju objekata, od modela R-CNN, preko Fast i Faster R-CNN-a do Mask R-CNN-a.

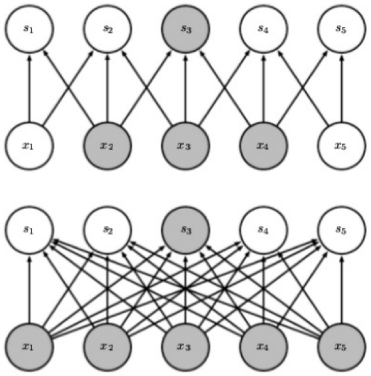
3.1. Konvolucijski sloj

Konvolucijski sloj je osnovni gradivni element konvolucijskih neuronskih mreža. Bitna razlika konvolucijskog i potpuno povezanog sloja je u tome što je kod potpuno povezanog sloja svaki neuron povezan sa svim neuronima prethodnog sloja, a kod konvolucijskog samo s malim dijelom (Slika 3.1). Dok bi potpuno povezani model ulaznu sliku promatrao u cjelini, konvolucije djeluju na djeliće slike i imaju mogućnost naučiti ključne značajke određene lokalnim svojstvima. Puno manji broj veza također znači i puno manje parametara, odnosno težina, što povlači i puno brže učenje i iskorištavanje modela. Učivi parametri nalaze se u jezgrama konvolucijskoj sloja. Jezgre su uglavnom malih prostornih dimenzija, dok dubinom odgovaraju broju kanala ulazne mape značajki. Broj kanala izlazne mape značajki ovisi o broju jezgri konvolucijskog sloja. Jezgre se pomiču za određen korak (eng. *stride*) po ulazu, te nad svakim djelićem mape obavljaju matematičku operaciju kako je ilustrirano na slici 3.2. Ako želimo zadržati prostorne dimenzije ulazne mape značajki, možemo koristiti nadopunu nulama (eng. padding) kako je ilustrirano na slici 3.3b.

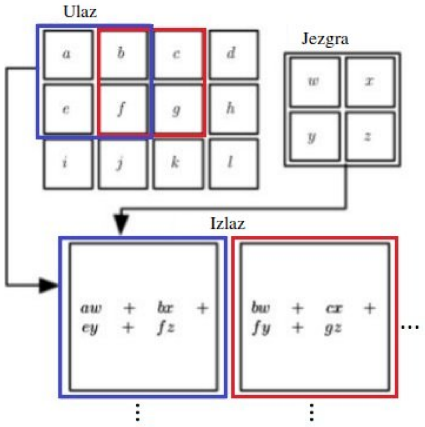
2D konvoluciju možemo opisati formulom:

$$q_{ij} = \sum_{u}^{k} \sum_{v}^{k} p_{i-o_k+u, j-o_k+v} \cdot w_{uv} \tag{3.1}$$

gdje je q izlazna mapa značajki, p ulazna mapa značajki, w jezgra, k veličina jezgre, a $o_k = \lfloor k/2 \rfloor + 1$.

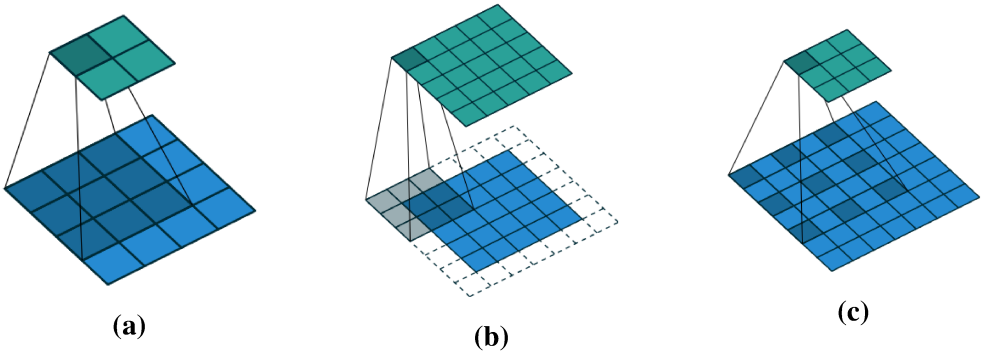


Slika 3.1: Usporedba konvolucijskog sloja (gore) i potpuno povezanog sloja (dolje) kako je prikazano u [27].



Slika 3.2: Ilustracija pomičnog prozora (jezgre) iz [27].

Receptivno polje značajke definiramo kao skup svih elemenata ulaznog sloja koje mogu utjecati na tu značajku. Možemo ga povećati povećanjem veličine jezgre, ali i dilatacijom odnosno širenjem jezgre. Konvolucija s dilatacijom 1 odgovara klasičnoj konvoluciji, dok je konvolucija s dilatacijom 2 ekvivalentna konvoluciji s većom jezgrom kod koje se svaki drugi redak i stupac sastoji od nula (Slika 3.3c).

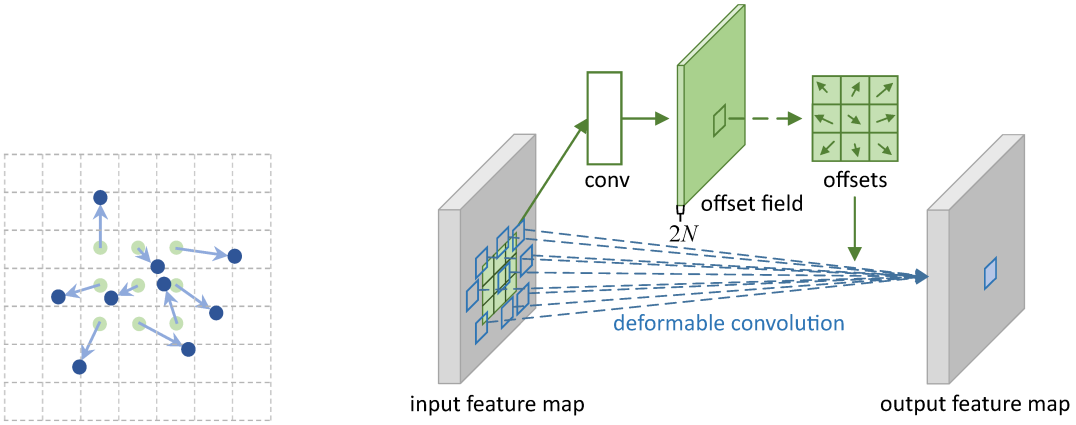


(a) **(b)** **(c)**

Slika 3.3: Na slikama su redom prikazane klasična konvolucija, konvolucija s nadopunjavanjem i dilatirana konvolucija, preuzeto iz [4].

3.2. Deformirajući konvolucijski sloj

Tradicionalni konvolucijski slojevi limitirani su fiksnom geometrijskom strukturom - pravokutnom mrežom. Glavna novost koju donose deformirajući konvolucijski slojevi je mogućnost pomicanja svake točke te mreže za određeni pomak koji se uči. Konvolucija tada djeluje na pomaknutim pozicijama. Na slici 3.4a ilustrirana je mreža veličine 7×7. Svjetlozelenom bojom prikazane su točke na koje bi djelovala standardna konvolucijska jezgre veličine 3×3, a plavom bojom ilustrirane su pomaknute točke na koje djeluje predložena deformirajuća konvolucija. Budući da pomaci nisu diskretni, vrijednosti se dobivaju bilinearnom interpolacijom.



(a) **(b)**

Slika 3.4: Na slici a) ilustrirana je mreža veličine 7×7. Svjetlozelenom bojom prikazane su točke na koje bi djelovala standardna konvolucijska jezgre veličine 3×3, a plavom bojom ilustrirane su pomaknute točke na koje djeluje predložena deformirajuća konvolucija. Na slici b) prikazana je arhitektura deformirajućeg konvolucijskog sloja, možemo uočiti dodatnu konvolucijsku granu kojom se uče pomaci, a djeluje na istoj ulaznoj mapi značajki kao i deformirajuća konvolucija. Ilustracije su preuzete iz [2].

Ako smo običnu dvodimenzionalnu konvoluciju opisali formulom 3.1, tada deformirajuću konvoluciju možemo izraziti formulom:

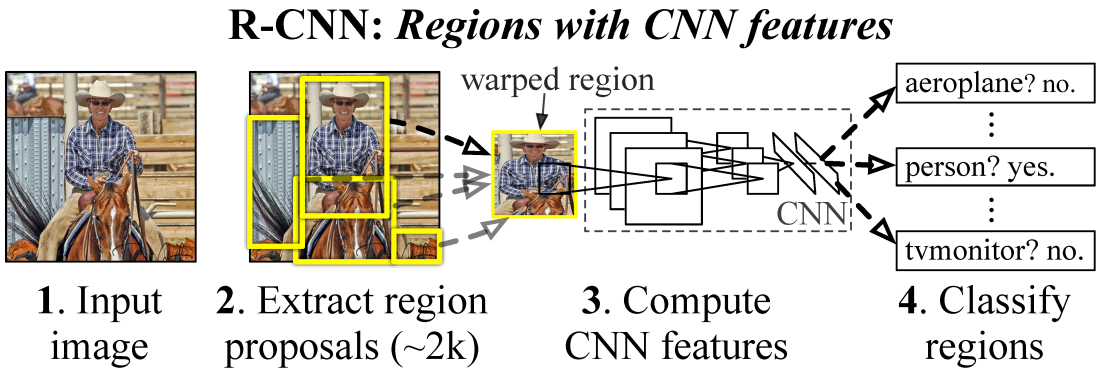
$$q_{ij} = \sum_{u}^{k} \sum_{v}^{k} p_{i-o_k+u+\Delta x, j-o_k+v+\Delta y} \cdot w_{uv} \tag{3.2}$$

gdje su Δx i Δy pomaci koje učimo paralelnom konvolucijskom granom na istoj ulaznoj mapi značajki kako je prikazano na slici 3.4b.

3.3. R-CNN

Detekcija objekata jedan je od osnovnih problema računalnog vida. Bilo da je riječ o ljudima, licima, automobilima ili tekstu detekcija je obično prvi korak ka rješavanju nekog složenijeg problema. Roos Girshick et al. u svojem radu [6] predstavili su model za detekciju objekata koji se kroz godine nadograđivao i na kojem se temelji mnoštvo modernih metoda. Osmislili su postupak u kojem se kombinira selektivno pretraživanje regija, konvolucijske neuronske mreže i metoda potpornih vektora (SVM). Postupak je ukratko opisan u nastavku:

1. Model predlaže najinteresantnije regije na ulaznoj slici (obično njih oko 2000).
2. Svaka od predloženih regija provlači se kroz konvolucijsku neuronsku mrežu i stvara se vektor značajki za pojedinu regiju.
3. Svaki vektor značajki iz prethodnog koraka klasificira se metodom potpornih vektora. Uz oznaku razreda, model na izlazu daje i 4 skalara koje označavaju odmake stranica opisnog okvira s ciljem postizanja veće preciznosti.
4. Provodi se suzbijanje nevažnih detekcija u dva koraka. Za svaki razred prolazimo kroz sve regije i ako dolazi do preklapanja (IoU veći od zadane konstante) uklanjamo onu regiju s manjom vjerojatnošću pronalaska objekta za taj razred, a zatim uklanjamo sve regije s vjerojatnošću manjom od 0.5.



Slika 3.5: R-CNN model prikazan u [6], sadrži sustav za pronalaženje interesnih regija, konvolucijsku neuronsku mrežu za izvlačenje značajki pojedine regije i SVM za klasifikaciju vektora značajki pojedine regije.

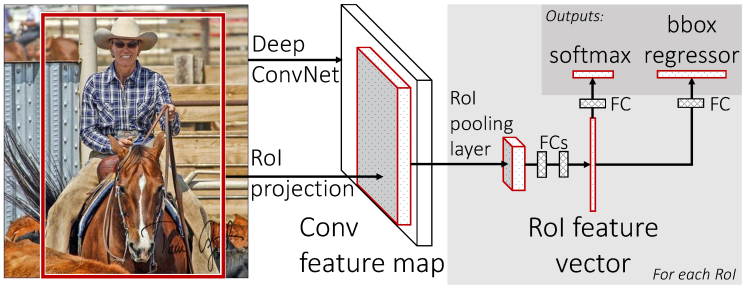
Ovakav pristup, iako vrlo značajan 2014. godine, imao je i brojne probleme:

1. U prvom koraku koristio se selektivni algoritam pretraživanja koji ne uči, stoga je upitna kvaliteta predlaganja interesnih regija.
2. Učenje je bilo teško pošto se R-CNN sastoji od dva dijela (CNN i SVM) koji se moraju zasebno učiti jedan za drugim.
3. Učenje je bilo sporo pošto je svaku od približno 2000 regija trebalo provući kroz CNN.
4. Evaluacija je bila spora, trajala je oko 47 sekundi na grafičkoj kartici i mreži VGG16.
5. Zahtjeva mnogo memorije za pohranu vektora značajku svake od približno 2000 interesnih regija.

3.4. Fast R-CNN

Autor modela R-CNN nešto kasnije predstavlja njegovo unaprijeđenje pod nazivom Fast R-CNN. Glavna prednost novoosmišljenog modela jest unificiranost svih dijelova klasičnog R-CNN-a u jedan sustav, što omogućuje puno jednostavnije učenje s kraja na kraj (eng. *end-to-end*). Novi algoritam čine sljedeći koraci:

1. Model predlaže najinteresantnije regije na ulaznoj slici (obično njih oko 2000).
2. Cijela ulazna slika provlači se kroz konvolucijsku neuronsku mrežu i stvara se mapa značajki.
3. Za svaku od predloženih regija se uzima odgovarajući dijelić mape značajki iz prethodnog koraka koji se onda provlači kroz sloj sažimanja (*RoI pooling*) i izravnava u vektor kako bismo za svaku regiju dobili pripadna vektor značajki fiksne veličine.
4. Tako dobiveni vektori značajki više ne odlaze u SVM, nego u potpuno povezanu neuronsku mrežu koja odrađuje posao klasifikacije i ugađanja opisnih okvira.



Slika 3.6: Fast R-CNN model prikazan u [5]. Cijela ulazna slika provlači se kroz konvolucijsku neuronsku mrežu i stvara se mapa značajki iz koje se zatim za svaku od predloženih regija uzima odgovarajući dijelić mape. SVM je zamjenjen potpuno povezanom neuronskom mrežom.

Ovakva arhitektura donosi brojne prednosti u odnosu na R-CNN:

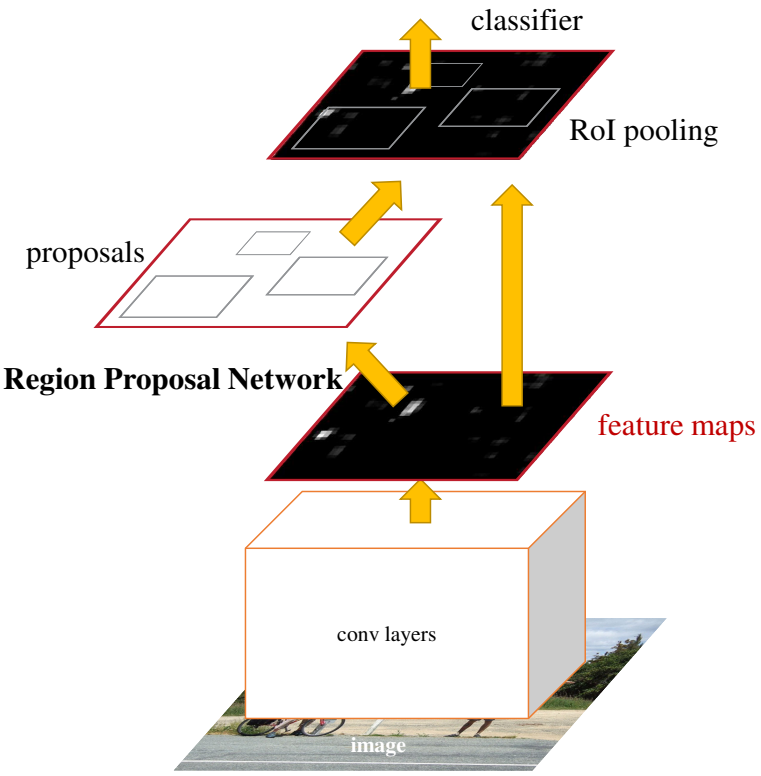
1. Kvalitetnije detekcije (veća srednja prosječna preciznost - mAP)
2. Učenje cijelog sustava istovremeno
3. Znatno (9 puta) kraće vrijeme učenja
4. Znatno (213 puta) kraće vrijeme evaluiranja
5. Eliminira potrebu za pohranom stotina gigabajta značajki na disku.

Fast R-CNN zadržava jedan bitan problem, a to je predlaganje regija selektivnim algoritmom pretraživanja koji ne uči.

3.5. Faster R-CNN

Ubrzo nakon Fast R-CNN-a, Shaoqing et al. predstavljaju još jedno poboljšanje koje je usmjereno na prvu fazu odnosno na predlaganje regija.

Umjesto sporog selektivnog pretraživanja koje ne uči, za pronalaženje regija koristi se duboka potpuno konvolucijska neuronska mreža kao što je prikazano na slici 3.7. Mreža za pronalaženje regija i mreža za klasifikaciju dijele određen broj zajedničkih konvolucijskih slojeva, što model čini još povoljnijim. Zaključno, model je brži od prethodnika, dok istovremeno predlaže kvalitetnije regije, što podiže njegovu preciznost.

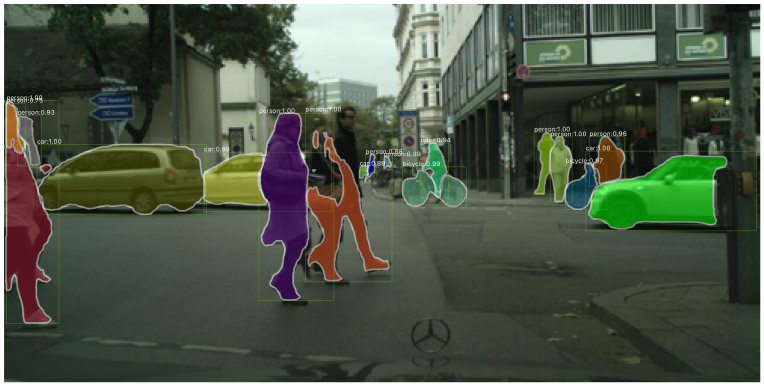


Slika 3.7: Faster R-CNN model prikazan u [21]. Za pronalaženje regija koristi se zasebna duboka konvolucijska neuronska mreža RPN. RPN na ulazu prima izlaznu mapu značajki posljednjeg konvolucijskog sloja ekstraktora značajki, a na izlazu daje niz pravokutnih regija uz mjeru pripadnosti regije nekom od objekata ili pozadini.

3.6. Mask R-CNN

U prethodnim potpoglavljima došli smo do efikasnog modela za detekciju objekata na slikama. Mask R-CNN ide korak dalje i objekte detektira na razini piksela, umjesto dotadašnjim opisnim okvirima. Model se nadograđuje na Faster R-CNN dodajući zasebnu neovisnu granu za predviđanje segmentacijske maske pojedinog objekta, što čini Mask R-CNN modelom za segmentaciju instanci.

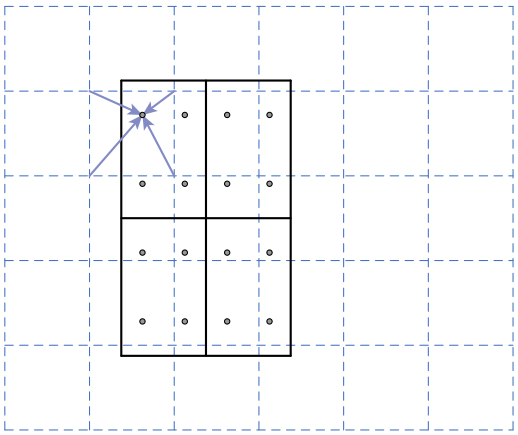
Nova grana za predviđanje segmentacijske maske zapravo je potpuno konvolucijska neuronska mreža koja se na pojedinu regiju primjenjuje paralelno s neuronskom mrežom zaduženom za klasifikaciju.



Slika 3.8: Primjer segmentacije instanci korištenjem modela Mask R-CNN na skupu Cityscapes prikazan u [9]. Dok bi semantička segmentacija sve objekte istog razreda (npr.sve automobile) svrstala u istu kategoriju, segmentacija instanci prepoznaje pojedini objekt.

Osim dodatne grane, bitna razlika između Faster i Mask R-CNN-a je i u načinu pronalaženja odgovarajućih značajki pojedinih regija te njihovom sažimanju. *RoI Pooling* je sloj koji se koristi u [5] za potrebe sažimanja pojedine regije u značajke fiksnih dimenzija. Pri određivanju značajki pojedine regije događaju se dvije kvantizacije koje dovode do nepreciznosti. Prva kvantizacija događa se prilikom određivanja odgovarajućih koordinata interesne regije na mapi značajki koja je S puta manja od izvorne slike, gdje dolazi do djeljenja i zaokruživanja $x' = \lfloor \frac{x}{S} \rfloor$ i $y' = \lfloor \frac{y}{S} \rfloor$. Do druge kvantizacije dolazi prilikom podjele odgovarajućeg isječka mape značajki na manje regije, tj. podjelom značajki prostornih dimenzija $h \times w$ na $N \times N$ jednakih regija čije su dimenzije onda $\lfloor \frac{h}{N} \rfloor \times \lfloor \frac{w}{N} \rfloor$. Kod modela Fast R-CNN ove nepreciznosti nisu imale znatan utjecaj na performanse, no kod modela Mask R-CNN eliminiranje ovakvih kvantizacija vrlo je bitno jer je preciznost važna prilikom određivanja segmentacijske maske instanci na razini piksela.

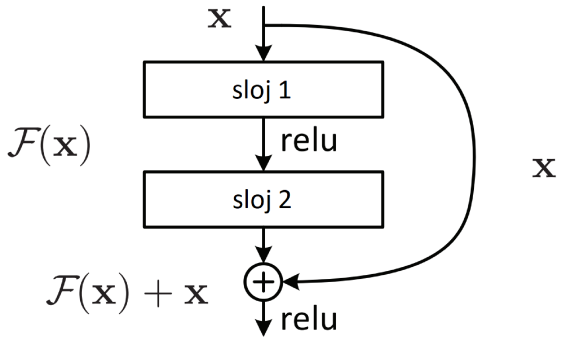
Roi Align je metoda kojom se izbjegavaju opisane kvantizacije izbjegavanjem zaokruživanja i korištenjem bilinearne interpolacije kao što je opisano na slici 3.9.
Mask R-CNN je i dalje jednostavan za učenje, te ne povećava značajno zahtjevnost izvođenja u odnosu na Faster R-CNN.



Slika 3.9: *RoI Align* iz [9]. Iscrtkana rešetka predstavlja mapu značajki, dok je punom linijom prikazana interesna regija podjeljena na 4 manje subregije. Unutar svake subregije se vrijednosti računaju na 4 mjesta bilinearnom interpolacijom na temelju 4 najbliže vrijednosti mape značajki, te se kao finalna vrijednost subregije uzima njihova prosječna ili maksimalna vrijednost.

4. Rezidualne neuronske mreže

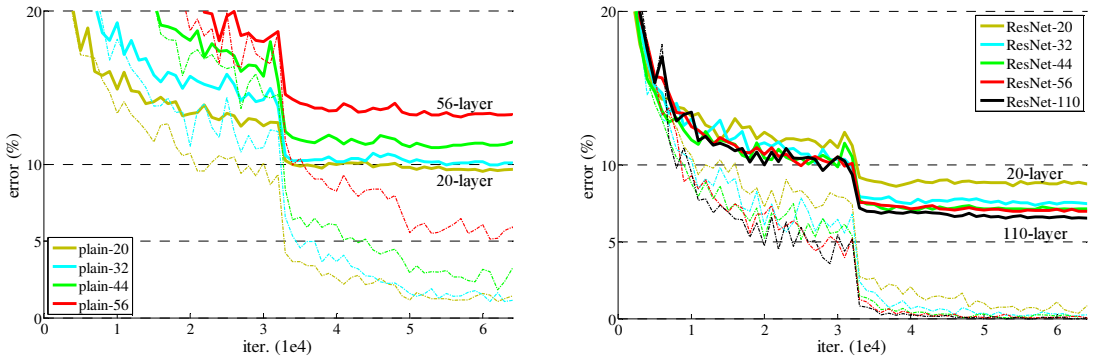
Duboke konvolucijske neuronske mreže postepeno uče značajke od jednostavnijih ka složenijima. Na primjer, ako imamo mrežu za klasifikaciju životinja, prvi sloj će naučiti pronalaziti jednostavne stvari poput rubova, neki sljedeći sloj prepoznavat će malo složenije stvari poput jednostavnih oblika ili tekstura, a još dublji sloj prepoznavat će elemente više razine poput lica. Iako veći broj slojeva doprinosi performansama, pokazalo se da dodavanje prevelikog broja slojeva ima negativan utjecaj. Rezidualne neuronske mreže rješavaju taj problem. Motivacija je sljedeća: Zamislimo mrežu A koja ima određenu grešku pri učenju. Konstruirajmo mrežu B tako da na mrežu A dodamo još slojeve X i Y, ali tako da ne mijenjaju izlaz iz mreže A. Očekivali bismo da mreža B nema veću grešku pri učenju od mreže A, no to nije slučaj. To pokazuje da slojevi X i Y utječu na mrežu, iako je njihova uloga prepisivanje podataka sa svog ulaza na izlaz, što znači da dubokim slojevima nije lagano naučiti funkciju identiteta. Element specifičan za rezidualne mreže je rezidualni blok prikazan na slici 4.1.



Slika 4.1: Rezidualni blok prikazan u [8].

Rezidualni blok rezultatu na izlazu proizvoljnog broja slojeva prije aktivacijske funkcije pridodaje ono što je u mreži naučeno prije tih slojeva. Ako slojevi ne doprinose mreži, ovime postižemo da joj neće niti pretjerano naštetiti. Konkretno, ako slojevi 1 i 2 na slici 4.1 imaju težine i pomake 0, glumit će funkciju identiteta pa neće negativno utjecati na performanse mreže. Dakle, rezidualnim neuronskim mrežama

lakše je naučiti funkciju identiteta nego klasičnim neuronskim mrežama. No, ni kod njih ne možemo dodavati iznimno velik broj slojeva, jer će primjerice ResNet-1202 imati gore performanse od ResNet-110 [8]. Zaključno, poanta nije samo da dodatni slojevi ne štete performansama mreže, već im i doprinose. Usporedba performansi klasičnih i rezidualnih mreža na skupu CIFAR-10 [13] prikazana je na slici 4.2.

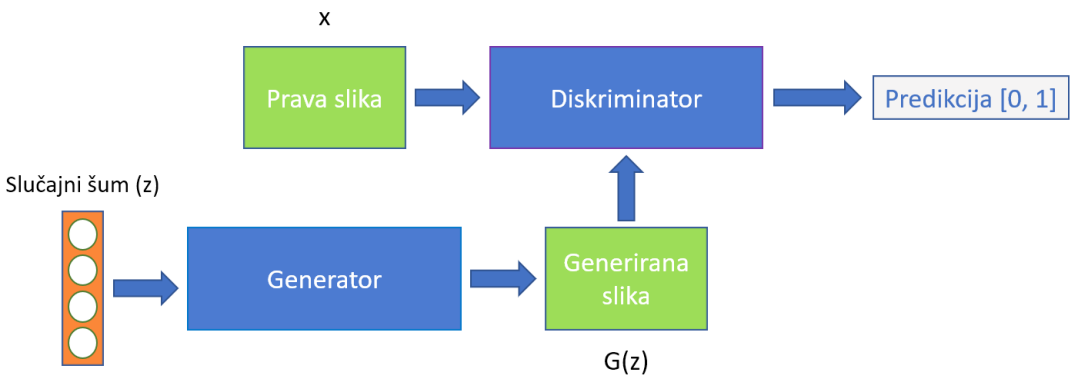


Slika 4.2: Usporedba performansi iz [8] na skupu CIFAR-10. Tanke linije označavaju greške na skupu za učenje, a debele linije označavaju greške na skupu za testiranje. Kod klasične neuronske mreže dodavanjem novih slojeva nakon dvadesetog performanse padaju, a kod rezidualne rastu.

Za potrebe ovog rada, po uzoru na [20], koristimo rezidualnu neuronsku mrežu s 18 slojeva (ResNet-18) kao ekstraktor značajki.

5. Generativni suparnički modeli

Generativni suparnički modeli [7] sastoje se od dvije neuronske mreže, generatora i diskriminatora. Svaka od dviju mreža ima svoj zadatak i gubitak koji nastoji minimizirati. Cilj generatora je proizvoditi što realnije uzorke, dok je cilj diskriminatora donesti sud je li ulazni uzorak stvaran (iz skupa za učenje) ili umjetan (proizveden od strane generatora). Rezultat ovakvog načina njihovog rada jest natjecanje koje ih tjera da budu sve bolji i bolji u svojim nastojanjima, sve do trenutka kada generator generira kvalitetne umjetne uzorke koje diskriminator ne može razlikovati od pravih.



Slika 5.1: Generativni suparnički model.

Na slici 5.1 je skicirana interakcija generatora i diskriminatora. Definirajmo izlaz diskriminatora kao vjerojatnost da je ulazni uzorak iz skupa za učenje. Ako je na ulazu neka prava slika x iz skupa za učenje, želimo da rezultat diskriminatora $D(x)$ bude što bliže 1. S druge strane, ako slučajni šum označimo sa z, a sliku proizvedenu u generatoru kao $G(z)$, generator učimo tako da nastoji izlaz diskriminatora $D(G(z))$ približiti jedinici, dok je istovremeno cilj diskriminatora prepoznati generiranu sliku i kao rezultat $D(G(z))$ vratiti vrijednost što bliže nuli.

5.1. Funkcija cilja

U ranije skiciranom modelu možemo prepoznati dvije povratne veze, diskriminatora s uzorcima iz skupa za učenje i generatora s diskriminatorom.

Postupkom učenja optimiziramo funkciju cilja s obzirom na parametre obje mreže, a to možemo prikazati sljedećim izrazom kao minimax igru:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{5.1}$$

Gornji izraz možemo razdvojiti na funkcije gubitka diskriminatora i generatora:

$$J^{(D)} = -\mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] - \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{5.2}$$

$$J^{(G)} = -J^{(D)} \tag{5.3}$$

Možemo primijetiti da izraz 5.2 odgovara unakrsnoj entropiji predikcija diskriminatora i oznaka binarnog razlikovanja stvarnih podataka od lažnih. Ako za primjer uzmemo klasičnu klasifikaciju uz korištenje unakrsne entropije, gradijenti padaju na nulu tek kada je mreža savršeno naučena i gradijent nam ionako više nije potreban. U slučaju generativnih suparničkih modela izuzetno dobar diskriminator također dovodi do nestanka gradijenata i to predstavlja problem jer je njegov gradijent i dalje potreban za učenje generatora.

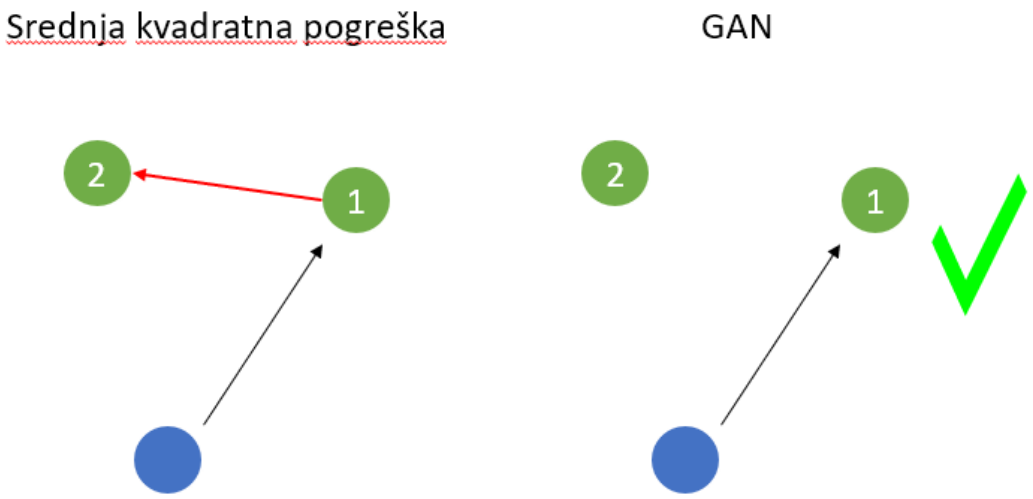
Ako se prisjetimo uvodnog dijela ovog poglavlja, rekli smo da generator nastoji izlaz diskriminatora $D(G(z))$ približiti nuli. Na taj način možemo i napisati funkciju gubitka generatora i time riješiti problem nestajućih gradijenata. Dobivene funkcije gubitka sada su:

$$J^{(D)} = -\mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] - \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{5.4}$$

$$J^{(G)} = -\mathbb{E}_{z\sim p_z(z)}[\log(D(G(z)))] \tag{5.5}$$

5.2. Usporedba s drugim pristupima

Kod generativnih modela cilj nam je generirati različite nove uzorke. Generativne suparničke mreže su veoma dobre upravo u tim slučajevima kada imamo više "točnih odgovora" odnosno kada smijemo generirati više različitih uzoraka, a jedini uvjet jest da izgledaju realistično. Navedeno je prikazano na intuitivnoj razini na slici 5.2.

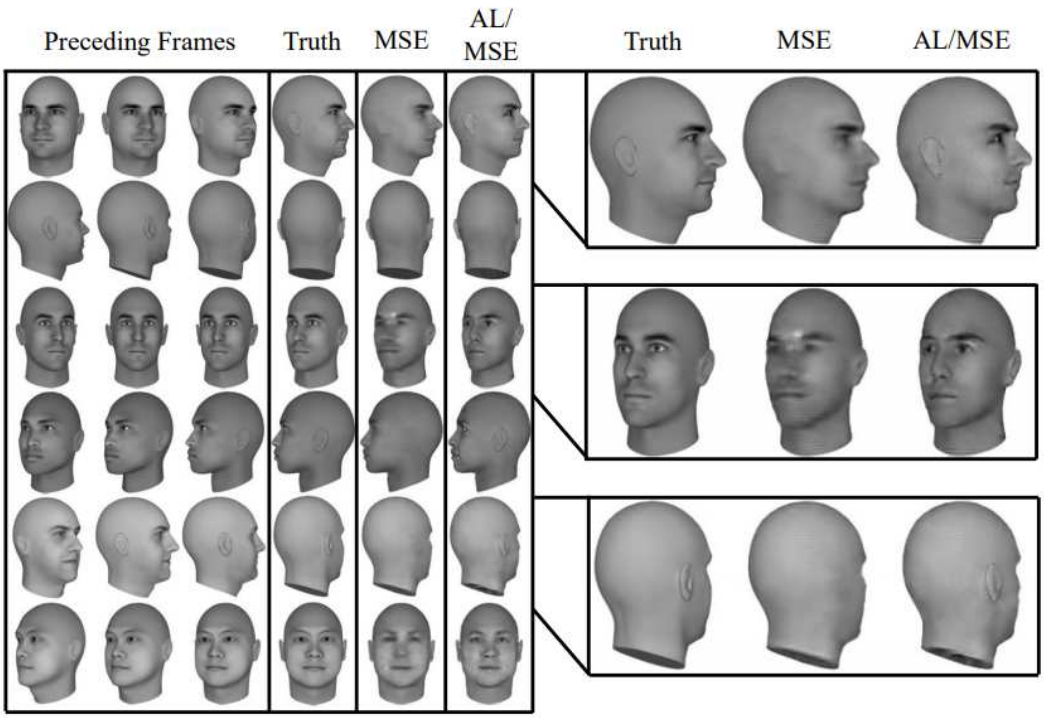


Slika 5.2: Ilustracija - usporedba srednje kvadratne pogreške i GAN-a.

Na ilustraciji su prikazana dva točna rješenja. Pretpostavimo da je zelena točka broj 2 bila priložena oznaka za neki ulaz (plava točka), a naš model je kao rezultat dao jednako dobar rezultat označen zelenom točkom broj 1. Ako koristimo srednju kvadratnu pogrešku, imat ćemo određeni gubitak ilustriran crvenom linijom, jer naša funkcija gubitka ne prepoznaje drugo točno rješenje. Nakon određenog vremena učenja, za određeni ulaz model će izbaciti srednju vrijednost svih točnih rezultata. Na slici 5.3 vidimo kako to izgleda kod predviđanja sljedećeg okvira videozapisa. U gornjem desnom kutu možemo na primjeru uha, koje se može nalaziti na više bliskih pozicija, kako model učen srednjom kvadratnom pogreškom ne zna točnu lokaciju uha te na izlazu daje srednju vrijednost svih mogućih, što se očitava kao mutna slika u usporedbi s puno bistrijom slikom generiranom suparničkim modelom.

U kontekstu ovog rada, generativne modele koristimo zato što želimo nešto slobodniji pristup modela kod predviđanja budućnosti. Nastojimo modelu reći da nije samo jedna mogućnost moguća i ohrabriti ga da riskira. Htjeli bi da naš model prilikom predviđanja budućeg stanja u obzir uzima sve moguće scenarije. Na primjer, umjesto da model odlučuje konzervativno i udove spaja jer tako minimizira gubitak, uvodimo

drugačije gubitke koji će modelu poručiti da vrijedi riskirati i pokušati pogoditi njihov budući položaj.

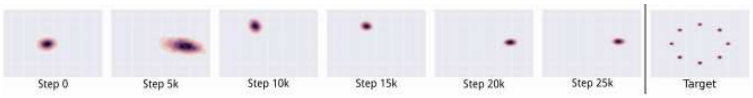


Slika 5.3: Predviđanje sljedećeg okvira za video rotirajućeg lica uz korištenje srednje kvadratne pogreške (MSE) i suprotstavljenih mreža (AL), prikazano u [16].

5.3. Problemi kod učenja generativnih modela

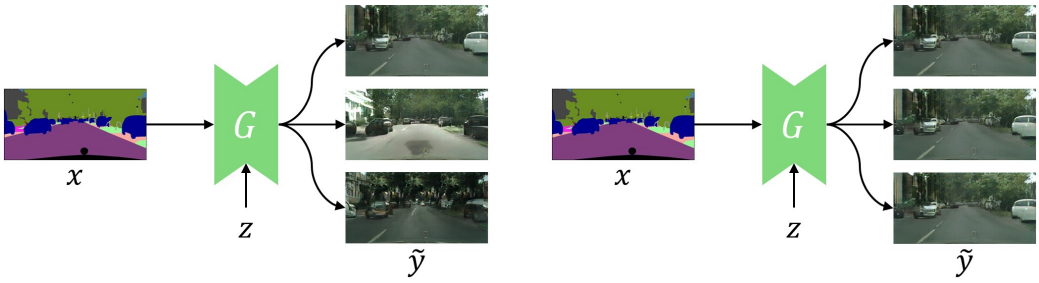
Najveći problem generativnim suparničkih modela je taj što postupak učenja ne konvergira uvijek. U ovom pristupu nemamo jednu funkciju gubitka koju minimiziramo, već je osnovno obilježje suparničkih modela to da imamo suparničke mreže od kojih svaka nastoji minimizirati svoju funkciju gubitka. U tako postavljenom problemu postoji šansa da nikada ne konvergiramo u ekvilibrij natjecanja dvaju mreža. Jedan oblik nekonvergencije koji se javlja kod generativnih suparničkih modela naziva se kolaps modova (eng. *mode collapse*). Kolaps modova označava situaciju u kojoj će generator pronaći slučaj na koji diskriminator loše reagira i posljedično sve svoje buduće generirane uzorke koncentrirati oko tog slučaja. S vremenom će diskriminator naučiti reagirati na taj konkretan slučaj, ali problem je što generator nakon toga neće naučiti da mora generirati raznolike uzorke, nego će pronaći novi slučaj na koji diskriminator loše reagira i dalje nove uzorke koncentrirati oko njega. Na slici 5.4 to je prikazano grafički.

Jedan od uzroka tog problema je neuravnoteženost učenja generatora i diskriminatora. Bitno je održavati balans prilikom učenja, jer ako diskriminator puno bolje obavlja svoj posao nego generator svoj, vraćat će vrijednosti toliko blizu nule ili jedinice da će generator imati problema s čitanjem gradijenata. S druge strane, ako je generator predobar iskorištavat će slabosti diskriminatora koje dovode do lažnih negativnih odluka (ulazna slika je prava). Češće je slučaj da diskriminator bolje obavlja svoj posao, stoga su razvijene tehnike kako bi se tom problemu doskočilo. Neke od njih su isključivanje neurona (eng. *dropout*) u diskriminatoru, pridodavanje šuma uzorku na ulazu diskriminatora, češće učenje generatora nego diskriminatora i korištenje drugačijeg optimizatora (npr. ADAM za generator, a SGD za diskriminator).



Slika 5.4: Na slici iz [18] prikazana je promjena distribucije generatora tijekom učenja.

Također, kako bismo riješili problem velike nestabilnosti kod učenja, autori modela Pix2Pix uz ranije navedenu funkciju cilja koriste i rekonstrukcijski, odnosno L1 ili L2, gubitak. To je izvedeno tako da se posao diskriminatora ne mijenja, dok generator mora zavarati diskriminator te dodatno generirati sliku sličnu pravoj (eng. *ground-truth*). Problem ovakvog pristupa je gubitak raznolikosti i generiranje veoma sličnih slika (slika 5.5), a rješenje će biti predstavljeno u poglavlju 7.

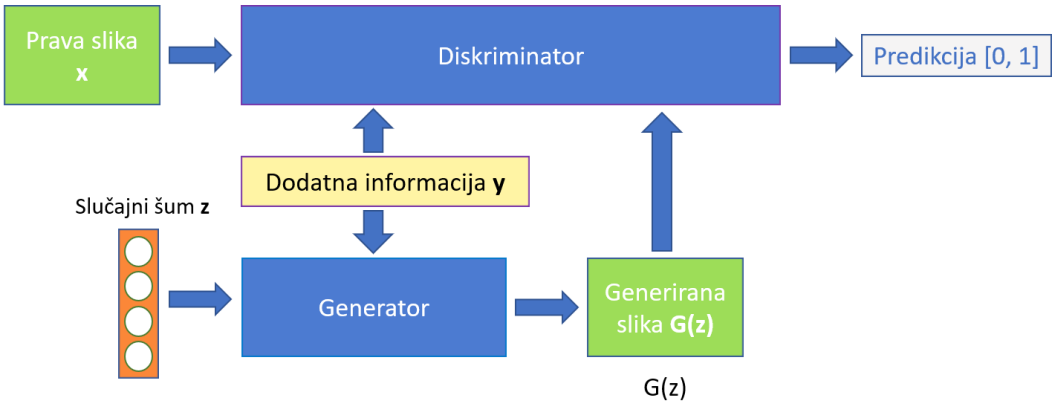


(a) *one-to-many* **(b)** *one-to-one* (Kolaps modova)

Slika 5.5: Usporedba željenog (lijevo) i dobivenog (desno) prilikom generiranja RGB slika iz njihovih semantičkih segmentacija, kako je prikazano u [22].

6. Uvjetni suparnički modeli

Generativne suparničke mreže može se proširiti na uvjetni suparnički model [19] ako su i generator i diskriminator uvjetovani nekom dodatnom informacijom y. y može biti bilo koja vrsta pomoćne informacije, jednostavna kao što je oznaka razreda ili složenija poput slike. Kondicioniranje možemo izvesti dodavanjem y kao dodatni ulaz u generator i diskriminator.



Slika 6.1: Uvjetni suparnički model.

Funkciju cilja sada možemo pisati kao:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x,y)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z,y),y))]. \quad (6.1)$$

Ako u obzir uzmemo i gubitak rekonstrukcije iz prethodnog poglavlja, konačna funkcija gubitka je:

$$\mathcal{L} = \mathcal{L}_{GAN} + \lambda_{Rec}\mathcal{L}_{Rec}. \quad (6.2)$$

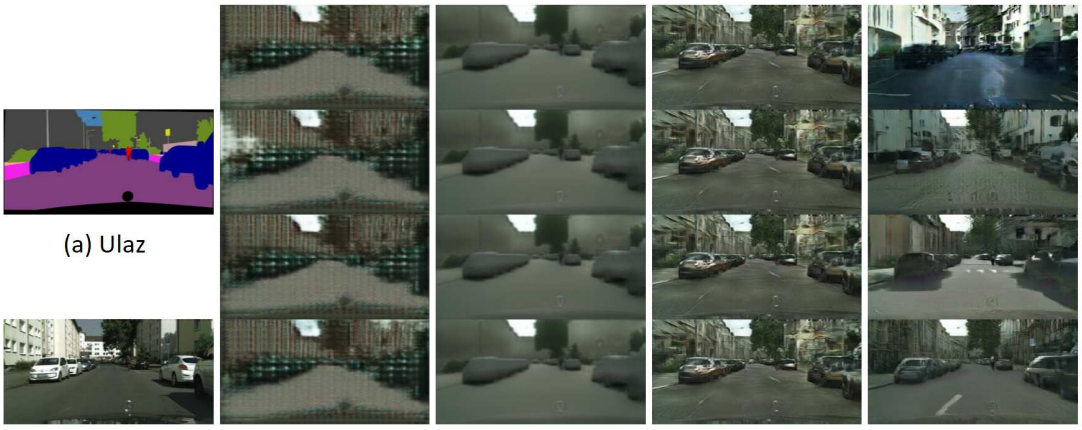
7. Multimodalno generiranje slika

Pokazalo se da je učenje generativnih modela poprilično nestabilno. Zbog toga se kod današnjih uvjetnih suparničkih modela u funkciju gubitka generatora uključuje L1 ili L2 udaljenost između generirane i stvarne slike kao rekonstrukcijska mjera koja prisiljava generator da proizvodi realistične slike slične pravima. Ipak, u [22] se pokazuje da je minimiziranje L2 gubitka ekvivalentno minimiziranju odstupanja i varijance predikcija. U kontekstu generativnih suparničkih modela, korištenje L2 rekonstrukcijskog gubitka prigušuje raznolikost generiranih uzoraka i dovodi do kolapsa modova.

Sochan Leei ostali u svojem radu [22] ponudili su rješenje na taj problem. Oni su definirali nove gubitke, koje nazivaju gubicima rekonstrukcije momenata (eng. *moment reconstruction losses*), i koriste ih umjesto klasičnih rekonstrukcijskih (L1/L2) gubitaka. S predloženim funkcijama gubitka postiže se i stabilnost pri učenju i raznolikost generiranih slika kao što je prikazano na slici 7.1.

Glavna ideja je koristiti procjenitelj najveće izglednosti kako bismo predvidjeli osnovna svojstva odnosno momente distribucije, konkretno očekivanje (moment prvog reda) i varijancu (moment drugog reda), odnosno srednju vrijednost i varijancu kod normalne distribucije. Odstupanje osnovnih svojstava distribucije pravog skupa i generiranog skupa podataka kažnjavamo. Autori su MLE gubitak za normalnu distribuciju definirali formulom 7.1.

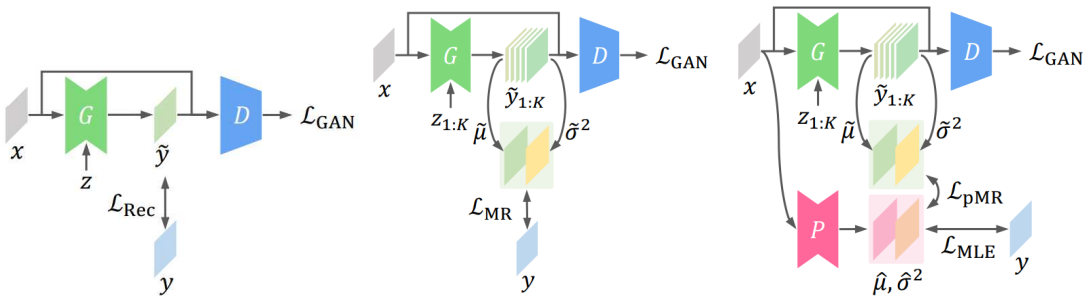
$$\mathcal{L}_{MLE,Gaussian} = \mathbb{E}_{x,y}\left[\frac{(y-\hat{\mu})^2}{2\hat{\sigma}^2} + \frac{1}{2}\log\hat{\sigma}^2\right], \text{ gdje je } (\hat{\mu}, \hat{\sigma}^2) = f_\theta(x) \qquad (7.1)$$



Slika 7.1: Usporedba performansi različitih modela preuzeta iz [22]. Na slici a) prikazan je ulaz u model, a na slici b) pravi izlaz (eng. *ground-truth*). U stupcu c) vidimo da je klasičan generativni suparnički model sam po sebi nestabilan, dok samostalno korištenje gubitka L1 proizvodi slike koje su slične pravima ali jako mutne i nerealistične kao što je prikazano u stupcu d). Kombinacija tih dvaju gubitaka dovodi do stabilnog učenja i realističnih slika, no pod cijenu raznolikosti - dobivene slike, prikazane u stupcu e), su skoro pa identične. U zadnjem stupcu prikazan je model naučen novopredloženim gubitkom, može se vidjeti da su dobivene slike i realistične i raznolike.

7.1. Gubici rekonstrukcije momenata

Prva funkcija gubitka nazvana je *Moment Reconstruction* (MR) gubitak, a uvjetni suparnički model koji ga koristi autori nazivaju MR-GAN. Na slici 7.2 prikazana je usporedba klasičnog modela na slici a) s MR-GAN-om na slici b).



(a) CGAN **(b)** MR-GAN **(c)** proxy MR-GAN

Slika 7.2: Usporedba arhitekture klasičnog uvjetnog suparničkog modela s predloženim modelima MR-GAN i *proxy* MR-GAN, preuzeto iz [22].

Uvođenjem nove funkcije gubitka MR nastaju dvije promjene:
1. Za svaku ulaznu sliku x generator proizvodi K različitih slika $\hat{y}_{1:K}$ uz pomoć K različitih mapa šuma $z_{1:K}$.
2. Funkcija gubitka primjenjuje se na momente (srednju vrijednost i varijancu) skupa generiranih uzoraka umjesto direktno na uzorcima.
Momente generirane distribucije procjenjujemo na sljedeći način:

$$\tilde{\mu} = \frac{1}{K}\sum_{i=1}^{K}\tilde{y}_i,\ \tilde{\sigma}^2 = \frac{1}{K-1}\sum_{i=1}^{K}(\tilde{y}_i - \tilde{\mu})^2,\ \text{gdje je } \tilde{y}_{1:K} = G(x, z_{1:K}). \tag{7.2}$$

Zatim računamo gubitak MR uvrštavanjem $\tilde{\mu}$ i $\tilde{\sigma}^2$ na mjesto $\hat{\mu}$ i $\hat{\sigma}^2$ u formuli 7.1. Tako dobiveni gubitak nazivat ćemo MR2, dok ćemo s MR1 označavati gubitak koji u obzir ne uzima varijancu. Kod gubitka MR1 izraz uvrštavamo $\tilde{\mu}$ na mjesto $\hat{\mu}$ u formuli:

$$\mathcal{L}_{MLE,Gaussian} = \mathbb{E}_{x,y}\left[(y - \hat{\mu})^2\right],\ \text{gdje je } \hat{\mu} = f_\theta(x). \tag{7.3}$$

Za stabilnije učenje, pogotovo u ranoj fazi, autori predlaži gubitak koji nazivaju *Proxy Moment Reconstruction* (*proxy* MR). Izmijenjeni model je prikazan na slici 7.2 c). Ključna razlika između gubitaka MR i *proxy* MR je uvođenje prediktora P, koji je zapravo kopija generatora bez ulaznog šuma. Prediktor se uči odvojeno i prije generatora, za funkciju gubitka koristi izraz 7.1, a uči momente $\hat{\mu}$ i $\hat{\sigma}^2$. Kasnije, pri učenju generatora, kao nit vodilju uzimamo upravo te momente koje je prediktor naučio. Izraz za gubitak *proxy* MR pišemo ovako:

$$\mathcal{L}_{pMR} = (\tilde{\mu} - \hat{\mu})^2 + (\tilde{\sigma}^2 - \hat{\sigma}^2)^2,\ \text{gdje je } (\hat{\mu}, \hat{\sigma}^2) = P(x). \tag{7.4}$$

Ovaj gubitak ćemo nazivati *proxy* MR2, dok ćemo kod gubitka *proxy* MR1 zanemariti varijancu, kao i kod običnog gubitka MR.

Metoda		Realističnost uzoraka	Raznolikost uzoraka
Nasumične prave slike		1.00	0.559
Pix2Pix + šum		$0.22^{\pm 0.04}$	0.004
BicycleGAN		$0.16^{\pm 0.03}$	0.191
Normalna distribucija	MR_1	$\mathbf{0.54}^{\pm 0.05}$	0.299
	MR_2	$0.14^{\pm 0.02}$	0.453
	proxy MR_1	$0.49^{\pm 0.05}$	**0.519**
	proxy MR_2	$0.44^{\pm 0.05}$	0.388
Laplacova distribucija	MR_1	$0.19^{\pm 0.03}$	0.393
	MR_2	$0.20^{\pm 0.04}$	0.384
	proxy MR_1	$0.19^{\pm 0.03}$	0.368
	proxy MR_2	$0.17^{\pm 0.02}$	0.380

Tablica 7.1: Usporedba performanse različitih funkcija gubitka, mjerenja su odrađena u sklopu [22] na modelu Pix2Pix i skupu za učenje Cityscapes. Raznolikost se ocjenjivala mjerom LPIPS [24], a realističnost se mjerila ručno od strane više ljudi kako je opisano u [22]. U obje mjere veće je bolje. Predloženi gubici MR i *proxy* MR imaju bolje performanse, pogotovo kad je u pitanju raznolikost.

Modeli MR-GAN i *proxy* MR-GAN oboje imaju svoje prednosti i mane. MR-GAN direktno pristupa pravoj slici y te se ne stvara pristranost zbog prediktora. S druge strane, korištenje prediktora pruža manju varijancu u ciljanim vrijednostima i dovodi do stabilnijeg učenja, pogotovo kad je broj generiranih uzoraka malen[1]. Također, kada koristimo *proxy* MR-GAN možemo uzeti prediktor s najmanjom greškom na skupu za validaciju kako bismo izbjegli prenaučenost. U tablici 7.1 vidimo da MR i *proxy* MR postižu približne rezultate na modelu Pix2Pix, pa ćemo u eksperimentima koristiti gubitke MR1 i MR2 radi jednostavnijeg učenja.

[1] Autori [22] za primjer daju K=12 na modelu SRGAN [14].

8. Predviđanje semantičke budućnosti iz konvolucijskih značajki

Većina prijašnjih radova s ciljem predviđanja budućnosti u video snimkama fokusirana je na predviđanje na razini RGB slika i naknadnoj segmentaciji. Uspjeh na tom području bio bi značajno postignuće zbog mogućnosti korištenja izrazito velikog skupa neoznačenih podataka za učenje. No, radovi unazad nekoliko godina pokazali su da je predviđanje puno efikasnije odraditi na semantičkoj razini (iz značajki u značajke), nego na RGB slikama. Uostalom, za probleme poput autonomne vožnje potrebno je da računalo prepoznaje okolinu na semantičkoj razini, pa je u tom slučaju predviđanje RGB budućnosti nepotrebna komplikacija.

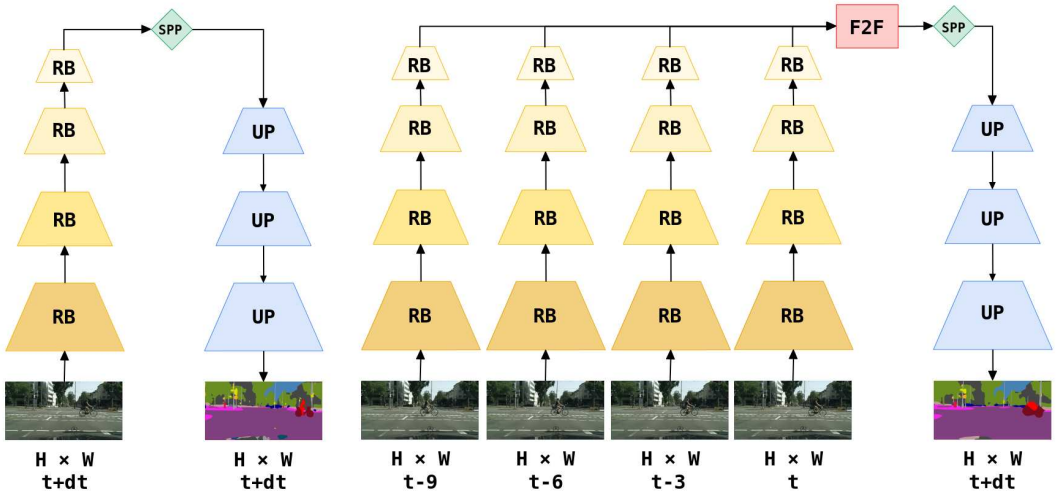
Kako su se mnogi tadašnji pokušaji na području predviđanja budućnosti iz značajki u značajke bazirali na semantičkoj segmentaciji, u [17] odlaze korak dalje pa semantičku budućnost predviđaju na razini instanci. Taj korak olakšava razumijevanje i predviđanje putanja kretanja pojedinih objekata. Predloženi model dijeli dobar dio arhitekture s modelom Mask R-CNN, uz dodatak predviđanja budućih okvira. Budući da je broj objekata na slikama varijabilan, ne predviđaju se direktno labele objekata, već konvolucijske značajke fiksnih dimenzija koje se zatim provlače kroz detekcijsku glavu i granu za naduzorkovanje (eng. *upsampling path*).

8.1. DeformF2F

Šarić i ostali u svojem radu [26] donose nekoliko bitnih novosti u odnosu na [17]:

1. Jedan *feature-to-feature* model (u nastavku F2F) koji radi na krajnjim, prostorno najmanjim, značajkama
2. Deformirajuće konvolucije umjesto klasičnih ili dilatiranih
3. Mogućnost finog ugađanja dva odvojeno učena podmodela (F2F i podsustav za segmentaciju koji čine ekstraktor značajki i grana za naduzorkovanje).

Njihov model postiže *state-of-the-art* performanse na srednjoročnoj (t+9) predikciji, te je drugi najbolji na kratkoročnoj (t+3) predikciji, što je značajan uspjeh uzevši u obzir jednostavnost i brzinu modela.



(a) **(b)**

Slika 8.1: Arhitektura modela iz [26]. Na slici a) je prikazan podsustav za semantičku segmentaciju koji se sastoji od grane za izvlačenje značajki i grane za naduzorkovanje. Na slici b) prikazan je kompletan model. Skroz lijevo ponovno se nalazi grana za izvlačenje značajki koja ovaj puta nezavisno djeluje na 4 ulazne slike, u sredini je crvenim pravokutnikom označen model za predviđanje budućih značajki F2F, dok je skroz desno ponovno prikazana grana za naduzorkovanje. U našoj arhitekturi SPP dolazi prije F2F.

Model je sačinjen od 3 dijela:

1. Grana za izvlačenje značajki
2. Model za predviđanje budućih značajki (F2F)
3. Grana za naduzorkovanje

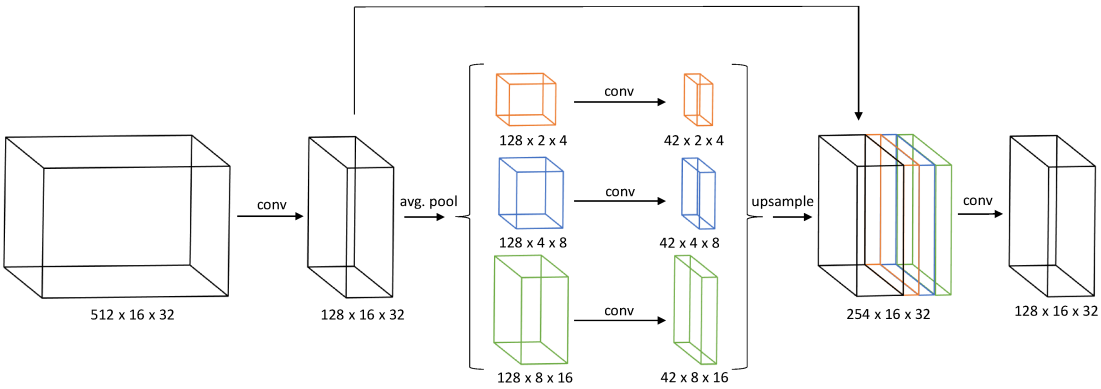
Arhitektura je ilustrirana na slici 8.1. Žuti trapezoidi predstavljaju rezidualne blokove koji formiraju sustav za izvlačenje značajki, dok je crvenim pravokutnikom označen F2F. F2F je glavni dio modela, zadužen za predviđanje budućih značajki. Plavim trapezoidima prikazani su moduli koji čine granu za naduzorkovanje, a zeleni romb označava prostorno piramidalno sažimanje. Na slici a) je prikazan model kojim se uči podsustav za semantičku segmentaciju, a sastoji se od grane za izvlačenje značajki i grana za naduzorkovanje. Također, ovaj model predstavlja *oracle* kojim predviđamo gornju granicu performansi modela tako da modelu na ulazu damo sliku "iz budućnosti" (onu čije značajke inače predviđa F2F), a na izlazu dobijemo njenu semantičku segmentaciju. Na slici b) je prikazan potpun model koji uz prethodno naučen podsustav za semantičku segmentaciju koristi i model za predviđanje budućih značajki F2F. Svi dijelovi modela pobliže su opisani u nastavku.

8.1.1. Grana za izvlačenje značajki

Rezidualna neuronska mreža Oslonac grane za izvlačenje značajki je mreža ResNet-18. Mreža je prednaučena na ImageNet [3] klasifikacijski, budući da se pokazalo da prijenosno učenje ima regularizacijsko svojstvo i općenito pozitivan utjecaj performanse. Mreža se sastoji od četiri bloka, od kojih svaki sadrži četiri konvolucijska sloja na istoj semantičkoj dimenziji, kojima prethode normalizacija grupe i ReLU aktivacija. Uz postepeno povećanje broja kanala, smanjuju se prostorne dimenzije, pa tako od početne dimenzije $3 \times 512 \times 1024$ dolazimo do $512 \times 16 \times 32$ što je detaljnije prikazano na slici 8.3.

Prostorno piramidalno sažimanje Na samom kraju nalazi se sloj prostornog piramidalnog sažimanja, u nastavku SPP (eng. *Spatial Pyramid Pooling*). Koristimo malo izmijenjeni SPP prigodan za korištenje u potpuno konvolucijskim neuronskim mrežama koji je prikazan na slici 8.2. Prosječnim sažimanjem ulaznih značajki prostornih dimenzija 16×32 na nekoliko razina dobivamo značajke 2, 4 i 8 puta manjih prostornih dimenzija. Dobivene značajke provlačimo kroz konvolucijski sloj (uz prethodnu

normalizaciju po grupi i ReLU aktivaciju), pa bilinearnom interpolacijom vraćamo na originalne dimenzije i pribrajamo ulaznim značajkama. Tako iz ulazne mape značajki sa 128 kanala i troje novodobivenih mapi značajki s 42 kanala dobivamo sjedinjenu mapu značajki s ukupno 254 kanala. Tu mapu zatim provlačimo kroz konvolucijski sloj s jezgrom veličine 1×1 kako bismo dobili naš standardi oblik u F2F prostoru, $128 \times 16 \times 32$. Ovaj prolaz ilustriran je na slici 8.3.



Slika 8.2: Sloj prostornog piramidalnog sažimanja (SPP).

8.1.2. Model za predviđanje budućih značajki F2F

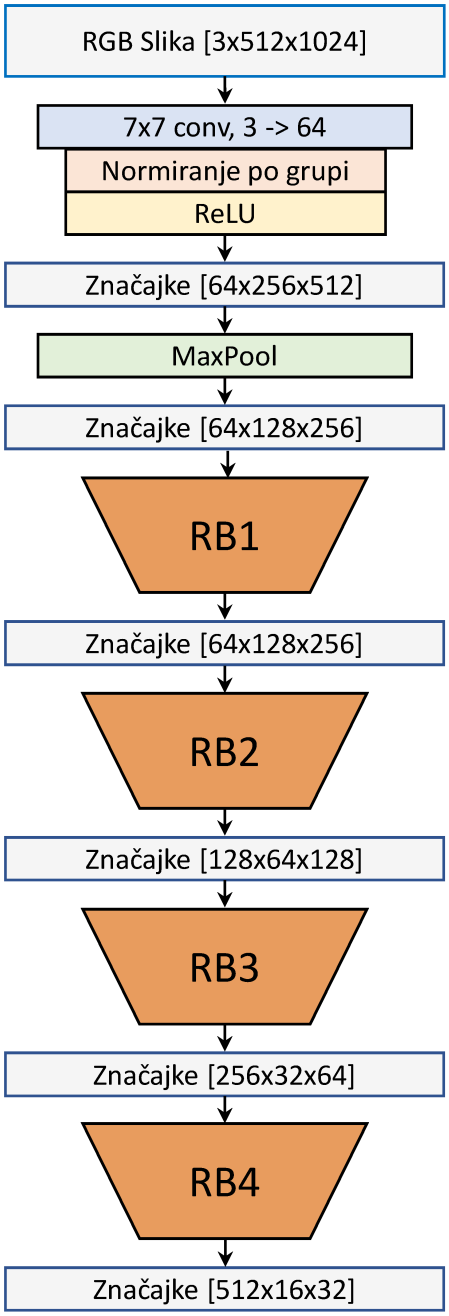
Prije prolaska kroz F2F, normaliziramo značajke na srednju vrijednost 0 i varijancu 1. Budući da smo ulazne značajke izračunali na četiri različite slike različitih trenutaka u prošlosti, mapu dimenzija $4 \times 128 \times 16 \times 32$ ćemo preoblikovati u $1 \times 512 \times 16 \times 32$, te ćemo dobivene značajke stopiti deformirajućim konvolucijskim slojem s jezgrom 1×1 u mapu dimenzija $1 \times 128 \times 16 \times 32$.

Zatim prolazimo kroz seriju od sedam deformirajućih konvolucija tijekom kojih se ni prostorne ni semantičke dimenzije ne mijenjaju. Ovaj niz deformirajućih konvolucija je glavni dio modela, zadužen za predviđanje budućih značajki. Budući da ulaz i izlaz modela žive u istom prostoru, model je moguće rekurzivno primjenjivati. Za potrebe ovog rada to ne radimo, no to svojstvo nam omogućuje nenadzirano učenje modela F2F.

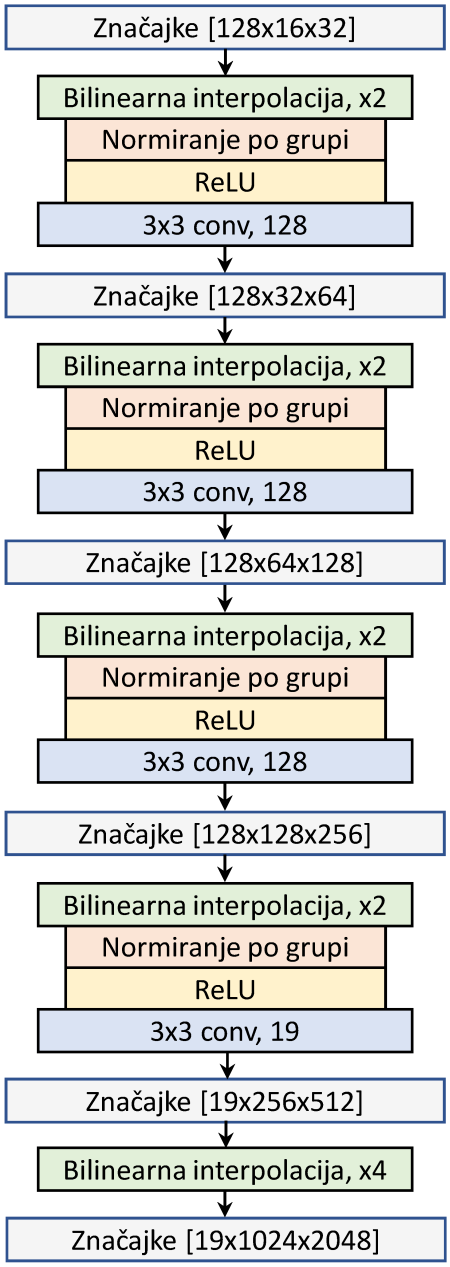
Na kraju, prije prolaska kroz granu za naduzorkovanje, značajke predikcija denormaliziramo.

8.1.3. Grana za naduzorkovanje

Grana za naduzorkovanje na ulazu prima značajke niske prostorne rezolucije koje se povećavaju prolaskom kroz blokove bilinearne interpolacije i konvolucije. Detalji prolaska opisani su slikom 8.4. Izmjena u odnosu na [20] je nedostatak lateralnih veza odnosno nedostatak pridodavanja značajki s različitih razina grane za izvlačenje značajki.



Slika 8.3: Grana za izvlačenje značajki.



Slika 8.4: Grana za naduzorkovanje.

9. Model i programska izvedba

Izgrađeni model kombinira prakse i tehnike prikazane u ranijim poglavljima: model za predviđanje budućnosti na razini značajki (F2F) uz minimalne intervencije preuzima ulogu generatora, prilagođeni PatchGAN ulogu diskriminatora, a u cilju postizanja što raznolikijih predikcija koriste se gubici MR1 i MR2.

9.1. Implementacija

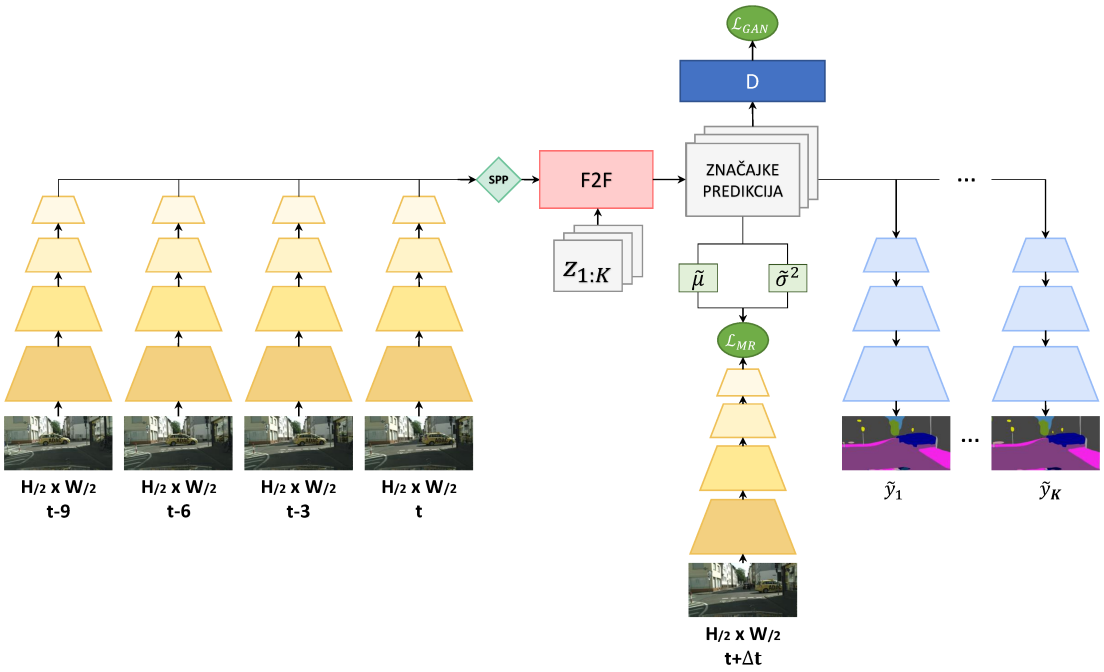
Implementacija ovoga rada je izravna nadogradnja na [26]. F2F je prilagođen za obavljanje funkcije generatora kako je opisano u 9.3.1, unaprijeđen je sustav za logiranje i vizualno praćenje napretka kroz epohe, te je implementiran diskriminator po uzoru na [10]. Sustavi za pretprocesiranje (izuzev prepolavljanja dimenzija ulaznih slika), izvlačenje značajki i naduzorkovanje nisu mijenjani. Za implementaciju korišten je programski jezik Python verzije 3.8.0, uz podršku raznih biblioteka od kojih je najbitnije spomenuti Pytorch, biblioteke PIL i cv2 za obradu slika te NumPy za matematičke izračune i manipulaciju tenzorima.

9.2. Podaci za učenje

Po uzoru na [26], za učenje i evaluaciju koristimo video sekvence iz skupa za učenje Cityscapes[1]. Skup sadrži 2975 scena (video sekvenci) za učenje, 500 za validaciju i 1525 za testiranje s oznakama za 19 razreda. Svaka scena opisana je s 30 slika, ukupnog trajanja 1.8 sekundi. Dakle, skup za učenje sadrži ukupno 150000 slika rezolucije 1024×2048. Semantička segmentacija (*ground-truth*) dostupna je za 20. sliku svake scene. Budući da je zbog uvođenja GAN-a arhitektura kompliciranija od [26], s ciljem smanjenja broja značajki i bržeg učenja sve slike iz skupa za učenje smo prvo prepolovili po širini i visini.

[1] Konkretno, koristi se leftImg8bit_sequence_trainvaltest

9.3. Arhitektura



Slika 9.1: Arhitektura našeg modela. Primijetimo sličnosti s osnovnim modelom na slici 8.1, ali i novosti u vidu šuma z, diskriminatora D, novih funkcija gubitka i većeg broja dobivenih predikcija.

9.3.1. Generator

Generator se bazira na modelu F2F predstavljenom u [26]. Kako bi predikcije na izlazu iz modela bile raznolike, uvodi se nasumičnost u obliku slučajnih tenzora normalne distribucije koji se pridodaju ulaznom tenzoru pri unaprijednom prolasku. Konkretno, šum koji ima 32 kanala i prostornim dimenzijama odgovara ulaznom tenzoru se unosi u posljednja dva konvolucijska sloja modela F2F. Isprobana su dva načina generiranja nasumičnosti.

Prvi način je da generiramo slučajan broj za svaku semantičku dimenziju i broj uzoraka koji treba generirati, pa se zatim dobiveni skalari proširuju na prostorne dimenzije $H \times W$. Time generiramo ukupno $B \cdot K \cdot N$ brojeva, gdje je B veličina mini-grupe, K broj uzoraka koje generiramo, a N broj kanala šuma.

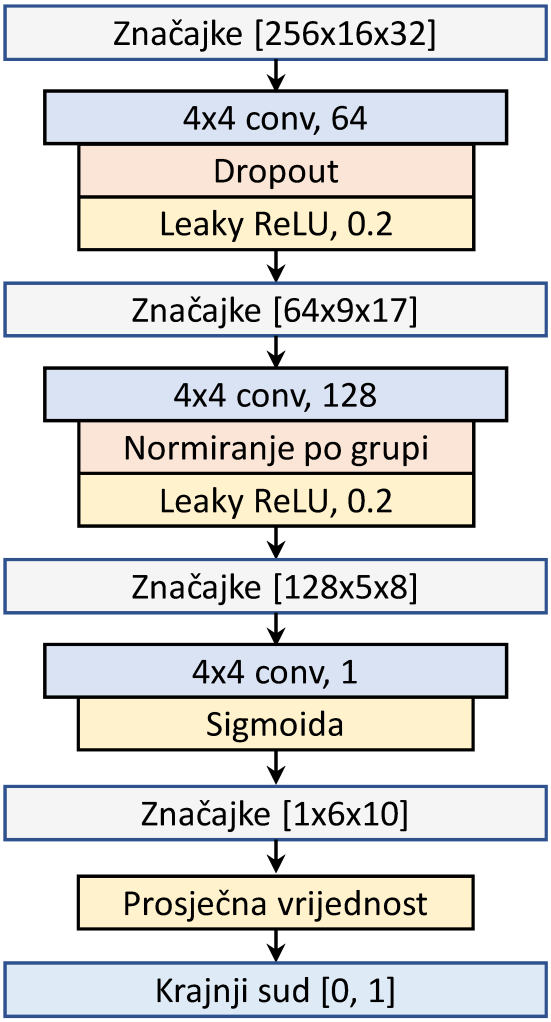
Drugi način je generiranje potpuno nasumične mape značajki odgovarajućih dimenzija uz zadani broj kanala, generiramo $B \cdot K \cdot N \cdot H \cdot W$ brojeva.

U oba slučaja se kanali latentne mape značajki z nadodaju na mapu značajki x koja se nalazi na ulazu u generator. Pokazalo se da drugi način daje bolje rezultate.

9.3.2. Diskriminator

Po prijedlogu iz [10], kao diskriminator se koristi PatchGAN [15]. Budući da naš PatchGAN na ulazu prima značajke manjih prostornih dimenzija, koristimo ga u izmijenjenom obliku s nešto manjim brojem konvolucijskih slojeva. Njegov posao je i dalje značajke svesti na manje regije, te su krajnje prostorne dimenzije značajki 6×10. Za svaku regiju donosi se sud pripada li ona skupu za učenje ili je proizvedena od strane generatora. Sudovi se zatim uprosječuju kako bismo na izlazu dobili konačnu odluku diskriminatora u obliku skalara.

Budući da je diskriminator previše dominirao prilikom učenja , uveden je *dropout* u prvom konvolucijskom sloju diskriminatora. Uglavnom gasimo između 50 i 65 posto značajki. Arhitektura je prikazana na slici 9.2.



Slika 9.2: Arhitektura diskriminatora.

9.4. Učenje modela

U potpoglavlju 9.2 opisan je korišteni skup za učenje i početna obrada ulaznih podataka. Ako trenutak u kojem se nalazimo označimo s t, onda kod kratkoročnog predviđanja koristimo značajke dobivene iz slika u trenucima t-9, t-6, t-3 i t kako bismo predvidjeli semantičku segmentaciju u trenutku t+3, odnosno u trenutku t+9 kod srednjoročnog predviđanja. Značajke jedne slike su prostornih dimenzija 16×32 i semantičke dimenzije 128.

Učenje možemo podijeliti na dva djela. Prvo gubitkom unakrsne entropije zajedno učimo granu za izvlačenje značajki i granu za naduzorkovanje, koji zapravo čine standardni model za semantičku segmentaciju [20, 12]. Zatim sve slike skupa za učenje koje se pri učenju mogu pronaći na ulazu provlačimo kroz granu za izvlačenje značajki i tako dobivene značajke spremamo na SSD. Time smo si uštedjeli vrijeme kod uzastopnog učenja i evaluiranja modela jer izbjegavamo prolaz kroz granu za izvlačenje značajki i umjesto toga učitavamo spremljene značajke. U drugom dijelu nenadzirano učimo model F2F.

Za razliku od [26], umjesto gubitka L2 koristimo gubitak MR i gubitak GAN-a. Rekonstrukcijskom gubitku dajemo nešto veći utjecaj ($\lambda_{MR} = 100$) nego suparničkom ($\lambda_{GAN} = 10$) kako bismo model pritegli na točne segmentacije. I za generator i za diskriminator koristimo optimizator Adam sa stopom učenja $4 \cdot 10^{-4}$ i koeficijentom za praćenje gradijenta 0.9 odnosno kvadrata gradijenta 0.99. Stopu za učenje smanjujemo kosinusnim kaljenjem (eng. *cosine annealing*) bez restarta do minimalne vrijednost $1 \cdot 10^{-7}$. Za vrijeme učenja generatora, on kreira 8 uzoraka pri unaprijednom prolasku.

Kako bismo balansirali rad generatora i diskriminatora u diskriminator uvodimo *dropout*. Na primjeru kratkoročnog učenja s gubitkom MR1 bez korištenja *dropouta* dolazi do stagniranja već oko četrdesete epohe, gdje je mIoU za 1.5 do 2 postotna boda manji od najboljih postignutih rezultata.

U nastavku je u grubim crtana prikazan postupak učenja generatora i diskriminatora na algoritmu 1.

Algoritam 1: Postupak učenja generatora i diskriminatora

Ulaz: Generator G, diskriminator, D

Ulaz: MR koeficijent λ_{MR}, GAN koeficijent λ_{GAN},

Ulaz: Ulazne značajke x, prava segmentacija y,

Ulaz: Broj predikcija koje treba generirati K,

Ulaz: Optimizatori generatora odnosno diskriminatora optimizator_G/D

dok *nije zadnja mini-grupa* **čini**

```
x, y = sljedeća_minigrupa()

# Učenje diskriminatora
optimizator_D.poništi_gradijente()
rezultat_stvarnih_slika = D.unaprijedni_prolaz(pridruži(x, y))
gubitak_stvarnih_slika =
        D.gubitak(rezultat_stvarnih_slika, stvarne_slike = Da)
generirane_slike = G.unaprijedni_prolaz(x)
rezultat_generiranih_slika=
        D.unaprijedni_prolaz(pridruži(x, generirane_slike))
gubitak_generiranih_slika =
        D.gubitak(rezultat_generiranih_slika, stvarne_slike = Ne)

ukupni_gubitak_D =
        λ_GAN * 0.5 * (gubitak_stvarnih_slika + gubitak_generiranih_slika)
ukupni_gubitak_D.prolaz_unatrag()
optimizator_D.korak()

# Učenje generatora
optimizator_G.poništi_gradijente()
generirane_slike_G = G.unaprijedni_prolaz(x, broj_predikcija=K)
parovi = pridruži(proširi(x, broj_predikcija=K), generirane_slike_G)
rezultat_generiranih_slika_G = D.unaprijedni_prolaz(parovi)
gubitak_MR = G.mr_gubitak(generirane_slike_G, y)
gubitak_GAN =
        G.gan_gubitak(rezultat_generiranih_slika_G, stvarne_slike = Da)
ukupni_gubitak_G = λ_MR * gubitak_MR + λ_GAN * gubitak_GAN
ukupni_gubitak_G.prolaz_unatrag()
optimizator_G.korak()
```

kraj

9.4.1. Dodatno - učenje generatora

U nastavku je prikazan nešto detaljniji pseudokod učenja generatora s MR1 (algoritam 2) i MR2 (algoritam 3) gubitkom po uzoru na [22].

Algoritam 2: Učenje generatora koristeći gubitak MR1

Ulaz: Generator G, diskriminator D,

Ulaz: MR koeficijent λ_{MR}, GAN koeficijent λ_{GAN},

Ulaz: Ulazne značajke x, prava segmentacija y,

Ulaz: Broj predikcija koje treba generirati K

za *i=1* ***do*** *K* **čini**

  $z_i \leftarrow$ GenerirajŠum()

  $\hat{y}_i \leftarrow$ G(x, z_i) {Generiraj predikciju}

kraj

$\hat{\mu} \leftarrow \frac{1}{K}\sum_{i=1}^{K}\hat{y}_i$ {Uzorkuj srednju vrijednost}

$\mathcal{L}_{MR} \leftarrow (y-\hat{\mu})^2$

$\mathcal{L}_{GAN} \leftarrow \frac{1}{K}\sum_{i=1}^{K} -\log D(x, y_i)$

$\theta_G \leftarrow$Adam$(\theta_G, \nabla_{\theta_G}[\lambda_{GAN}\mathcal{L}_{GAN} + \lambda_{MR}\mathcal{L}_{MR}])$

Algoritam 3: Učenje generatora koristeći gubitak MR2

Ulaz: Generator G, diskriminator D,

Ulaz: MR koeficijent λ_{MR}, GAN koeficijent λ_{GAN},

Ulaz: Ulazne značajke x, prava segmentacija y,

Ulaz: Broj predikcija koje treba generirati K

za *i=1* ***do*** *K* **čini**

  $z_i \leftarrow$ GenerirajŠum()

  $\hat{y}_i \leftarrow$ G(x, z_i) {Generiraj predikciju}

kraj

$\hat{\mu} \leftarrow \frac{1}{K}\sum_{i=1}^{K}\hat{y}_i$ {Uzorkuj srednju vrijednost}

$\hat{\sigma}^2 \leftarrow \frac{1}{K-1}\sum_{i=1} K(\hat{y}_i-\hat{\mu})^2$ {Uzorkuj varijancu}

$\mathcal{L}_{MR} \leftarrow \frac{(y-\hat{\mu})^2}{2\hat{\sigma}^2} + \frac{1}{2}\log\hat{\sigma}^2$

$\mathcal{L}_{GAN} \leftarrow \frac{1}{K}\sum_{i=1}^{K} -\log D(x, y_i)$

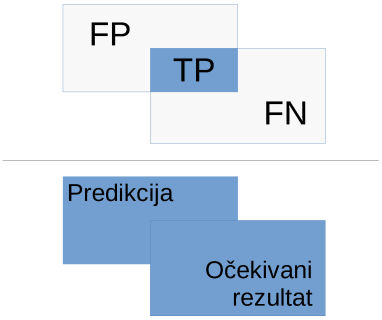
$\theta_G \leftarrow$Adam$(\theta_G, \nabla_{\theta_G}[\lambda_{GAN}\mathcal{L}_{GAN} + \lambda_{MR}\mathcal{L}_{MR}])$

10. Eksperimenti

10.1. Metrike

10.1.1. mIoU

Uspješnost modela ocjenjujemo Jaccardovim indeksom (eng. *Intersection over Union - IoU*). Preciznije, za svaki od 19 razreda računamo omjer presjeka i unije semantičke segmentacije dobivene na izlazu modela i prave segmentacije (eng. *ground-truth*), te uzimamo prosječnu vrijednost po razredima (eng. *mean Intersection over Union - mIoU*).



$$IoU = \frac{TP}{TP + FP + FN} \tag{10.1}$$

$$mIoU = \frac{1}{|C|} \sum_{C} IoU_C \tag{10.2}$$

Slika 10.1: IoU - omjer presjeka i unije.

Rezultati su opisani i metrikom mIoU-MO (*Moving Objects*), koja je zapravo mIoU mjeren na 8 razreda koje čine pomični objekti (ljudi, motoristi, automobili, kamioni, autobusi, vlakovi, motocikli i bicikli) kako bismo bolje ocijenili sposobnost modela da predviđa kretanje.

10.1.2. MSE

Srednju kvadratnu pogrešku (eng. *Mean Squared Error - MSE)* ovdje spominjemo u kontekstu usporedbe sličnosti dviju slika. Uspoređujemo kvadratnu udaljenost svakog piksela kao što je opisano formulom 10.3, gdje su I i K oznake dvaju slika, w njihova

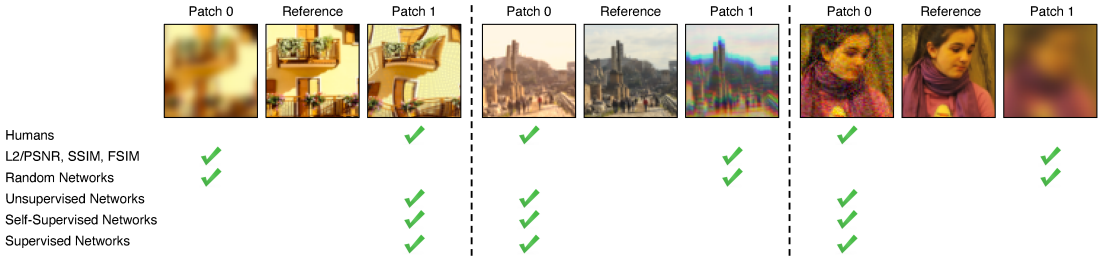
širina, a h visina.

$$MSE = \frac{1}{hw} \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} [I(i,j) - K(i,j)]^2 \tag{10.3}$$

Ako MSE iznosi 0, riječ je o identičnim slikama, a sve veća vrijednost označava sve različitije slike. Pokazuje se da srednja kvadratna pogreška općenito nije najbolja mjera za sličnost slika zbog preosjetljivosti na intenzitet piksela, pa tako mala promjena u kontrastu čini veliku razliku. Zbog toga se danas češće koristi složenija mjera SSIM [23]. Budući da ne generiramo RGB slike budućnosti, već samo njihovu semantičku segmentaciju, a SSIM je računski puno zahtjevniji od MSE, nećemo ga koristiti. Također, SSIM nije naročito informativan na ovom tipu predikcija, pa tako sličnost generiranih semantičkih segmentacija čija mjera MSE iznosi 3, SSIM ocjenjuje s 0.9999999999999996 (skoro identične).

10.1.3. LPIPS

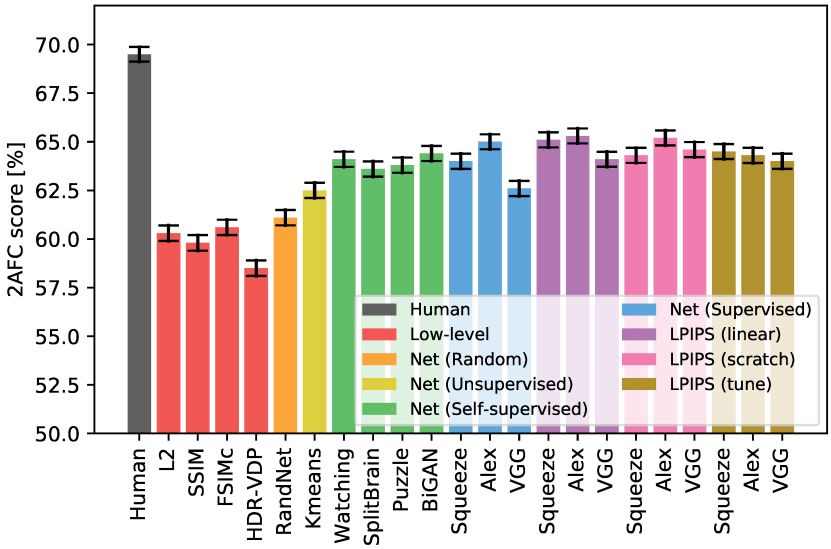
Po uzoru na [22], za kvantificiranje raznolikosti generiranih slika korištena je i mjera LPIPS [24]. LPIPS pokazuje da značajke dubokih modela mogu veoma dobro opisati sličnost dviju slika, puno bolje nego tradicionalne mjere poput L2 ili SSIM. Raznolikost se ocjenjuje prema razlikama u aktivacijama dubokih neurona. Pokazuje se da je takva mjera puno bliža ljudskom poimanju sličnosti dviju slika kao što vidimo na slici 10.2.



Slika 10.2: Na ova tri primjera iz [24] se najbolje vidi kako LPIPS pruža kvalitetniju procjenu sličnosti nego SSIM, L2 i slične mjere.

Pokazuje se da je mjerenje čak i na nasumično inicijaliziranim značajkama u nekim slučajevima bolja mjera sličnosti od tradicionalnih. Ipak, puno bolji rezultati se postižu na učenim modelima. Učenje se može obavljati na različitim zadacima, a pritom

nismo vezani ni za arhitekturu jer su autori pokazali približne performanse na različitim arhitekturama poput mreža VGG, AlexNet i SqueezeNet. Također, autori u [24] predstavljaju i specijalizirane metode učenja modela koje postižu još bolje rezultate.



Slika 10.3: Usporedba performansi različitih načina mjerenja sličnosti, preuzeto iz [24]. Sličnost je mjerena na izlaznim slikama konvolucijskih mreža na zadacima kao što je realistično podizanje rezolucije, interpolacija okvira, odmućivanje videa i bojanje slika.

Budući da naš model na izlazu daje semantičke segmentacije čija je struktura ograničena, pokazalo se da je mjera MSE sasvim dovoljna za ocjenu raznolikosti.

10.1.4. Vizualna ocjena

Osim brojčano izražene preciznosti i raznolikosti generiranih uzoraka, u nastavku će biti prikazani sami uzorci, te dvije sive slike izračunate na logitima dimenzija [Broj generiranih predikcija $\times$ broj razreda $\times$ visina $\times$ širina]:

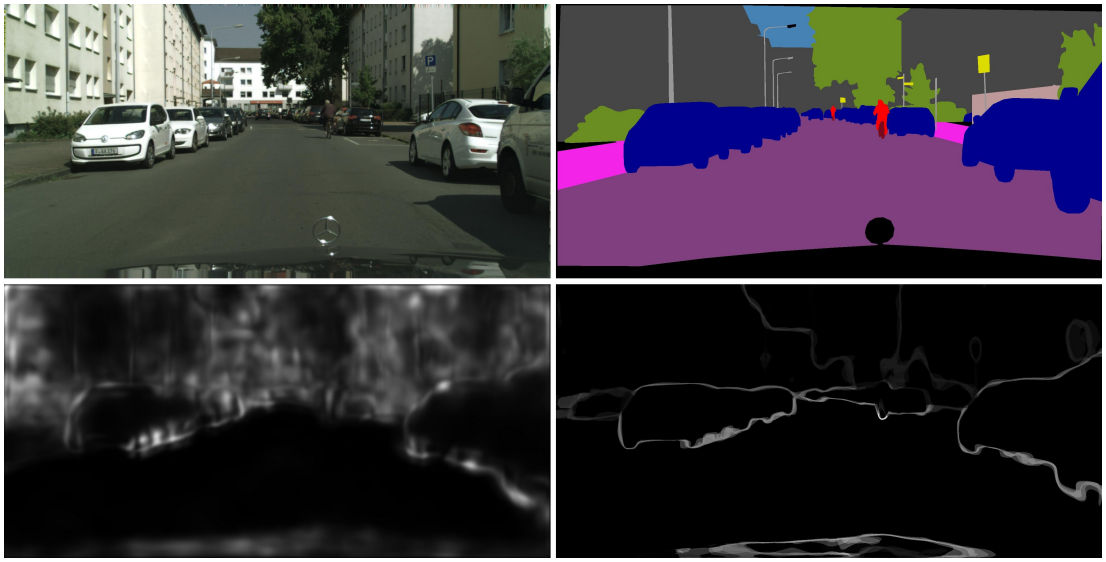
a) srednja varijanca logita po mapama značajki

```
logits.var(dim=0).mean(dim=0)
```

b) varijanca diskretnih predikcija

```
logits.argmax(dim=1).double().var(dim=0)
```

Primjer sivih slika prikazan je na figuri 10.4. Prva siva slika dočarava nam područja nesigurnosti, dok na drugoj vidimo područja koja su na različitim generiranim uzorcima svrstani u različite razrede.



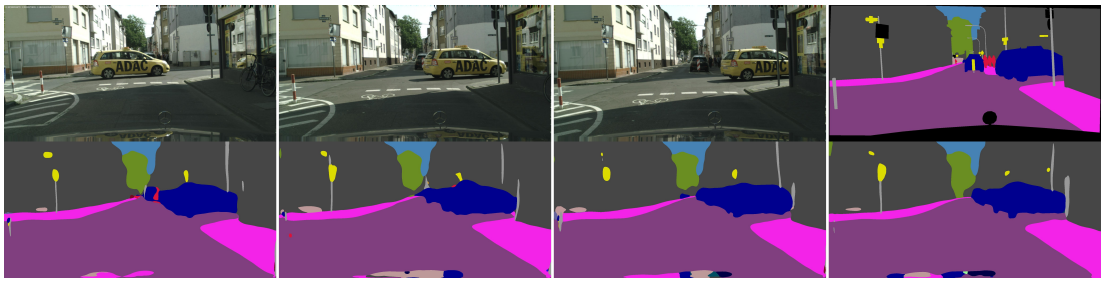
Slika 10.4:

Prvi red - na prvoj slici je prava RGB slika "iz budućnosti", a na drugoj slici je njena *ground-truth* segmentacija.

Drugi red - na prvoj slici je prikazana srednja varijanca logita po mapama značajki, što je nesigurnost modela na nekom području veća, to je područje bjelije. Na drugoj slici je prikazana varijancu diskretnih predikcija, što su područja više svrstana u različite razrede na različitim generiranim uzorcima, to su bjelija. U nastavku rada će biti prikazani generirani uzorci u kombinaciji sa sivim slikama.

10.2. Eksperimentalni rezultati

Proveli smo eksperimente na zadacima kratkoročnog i srednjoročnog predviđanja na modelima učenim jednakim hiperparametrima, izuzev postotka ugašenih neurona koji varira za različite zadatke i korištene funkcije gubitka. Korištenje gubitka MR1 rezultiralo je visokom ocjenom mIoU koja je usporediva s osnovnim modelom, a u nekim slučajevima i bolja. S druge strane, korištenje gubitka MR2 rezultiralo je padom kvalitete predikcija, a time i padom ocjene mIoU, no slike su puno raznovrsnije nego što je to slučaj s korištenjem gubitka MR1. Da mIoU nije jedina relevantna ocjena našeg modela najbolje možemo vidjeti na slici 10.5. Iako korištenje gubitka MR2 smanjuje mIoU za nekoliko postotnih poena, upravo korištenjem njega dobivamo najinteresantnije rezultate kod kratkoročnog predviđanja. Primijetimo da su na prvoj slici vidljivi ljudi, koje na zadnjoj slici koju je model vidio zaklanja automobil. U sljedećem trenutku automobil otkriva prostor iza sebe, a naš model po prvi puta na skoro pa točnom mjestu model predviđa ljude (drugi red, prva predikcija). Tako nešto nismo uspjeli postići s osnovnim modelom, pa čak ni s generativnim uz korištenje gubitka MR1. S gubitkom MR2 ovakve predikcije su moguće, ali još uvijek rijetke, pa smo i u ovom konkretnom slučaju ljude na pravom mjestu vidjeli samo u jednoj od dvanaest predikcija.



Slika 10.5: Predviđanje kratkoročne budućnosti s gubitkom MR2. U prvom redu su prikazani: prva i zadnja slika koje je model vidio (od ukupno četiri), slika iz budućnosti i njen *ground-truth*. U drugom redu prikazane su četiri od dvanaest dobivenih predikcija.

Glavni rezultati prikazani su u tablicama 10.1 i 10.2. Valja imati na umu da su ocjene prikazane u [26] dobivene na duplo većoj rezoluciji, stoga njihov model ponovno učimo na slikama duplo manje rezolucije 5 puta i uzimamo prosječne mjere mIoU i mIoU-MO. Rezultate našeg modela mjerimo na 5 evaluacija za svaki od 3 naučena modela za pojedini zadatak. Nešto bolju mIoU ocjenu našeg modela dobivamo ako dobivene predikcije usrednjimo, iako to pobija prvotnu zamisao rada.

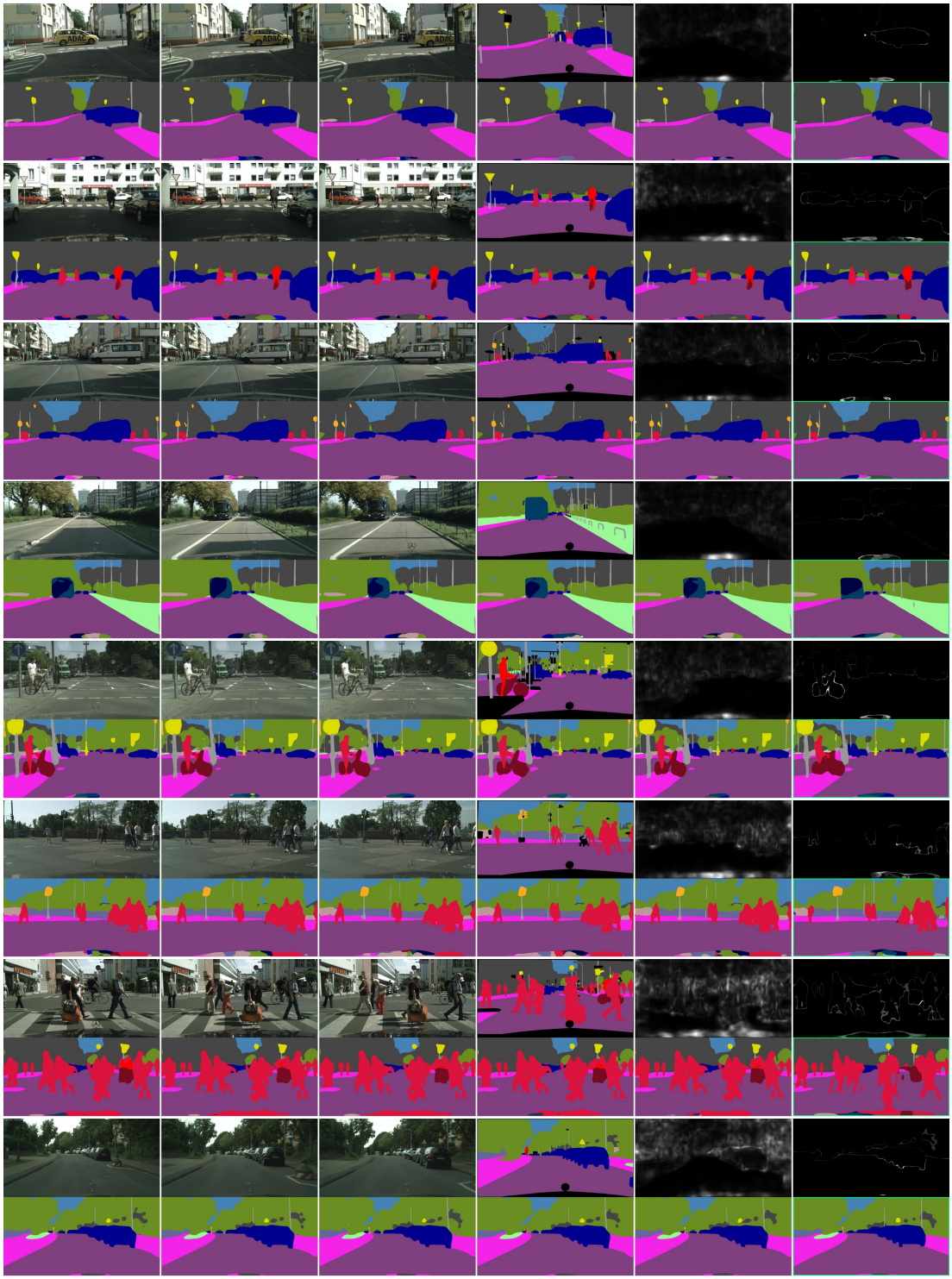
Metoda	Gubitak	mIoU	mIoU-MO	MSE	LPIPS
Oracle	L2	66.12	64.24	/	/
Segmentacija zadnje slike	L2	49.87	45.63	/	/
DeformF2F-8[26]	L2	$58.98^{\pm 0.17}$	$56.00^{\pm 0.20}$	/	/
MM-DeformF2F	MR1	$59.22^{\pm 0.05}$	$56.40^{\pm 0.08}$	$1.21^{\pm 0.05}$	0.0482
MM-DeformF2F usrednjen	MR1	$59.46^{\pm 0.06}$	$56.66^{\pm 0.09}$	/	/
MM-DeformF2F	MR2	$53.85^{\pm 0.35}$	$49.52^{\pm 0.64}$	$4.38^{\pm 0.24}$	0.1519
MM-DeformF2F usrednjen	MR2	$56.81^{\pm 0.20}$	$53.38^{\pm 0.43}$	/	/

Tablica 10.1: Rezultati kratkoročnog predviđanja. Uz korištenje gubitka MR1 naš model postiže nešto bolji mIoU, dok su uz gubitak MR2 predikcije puno raznovrsnije.

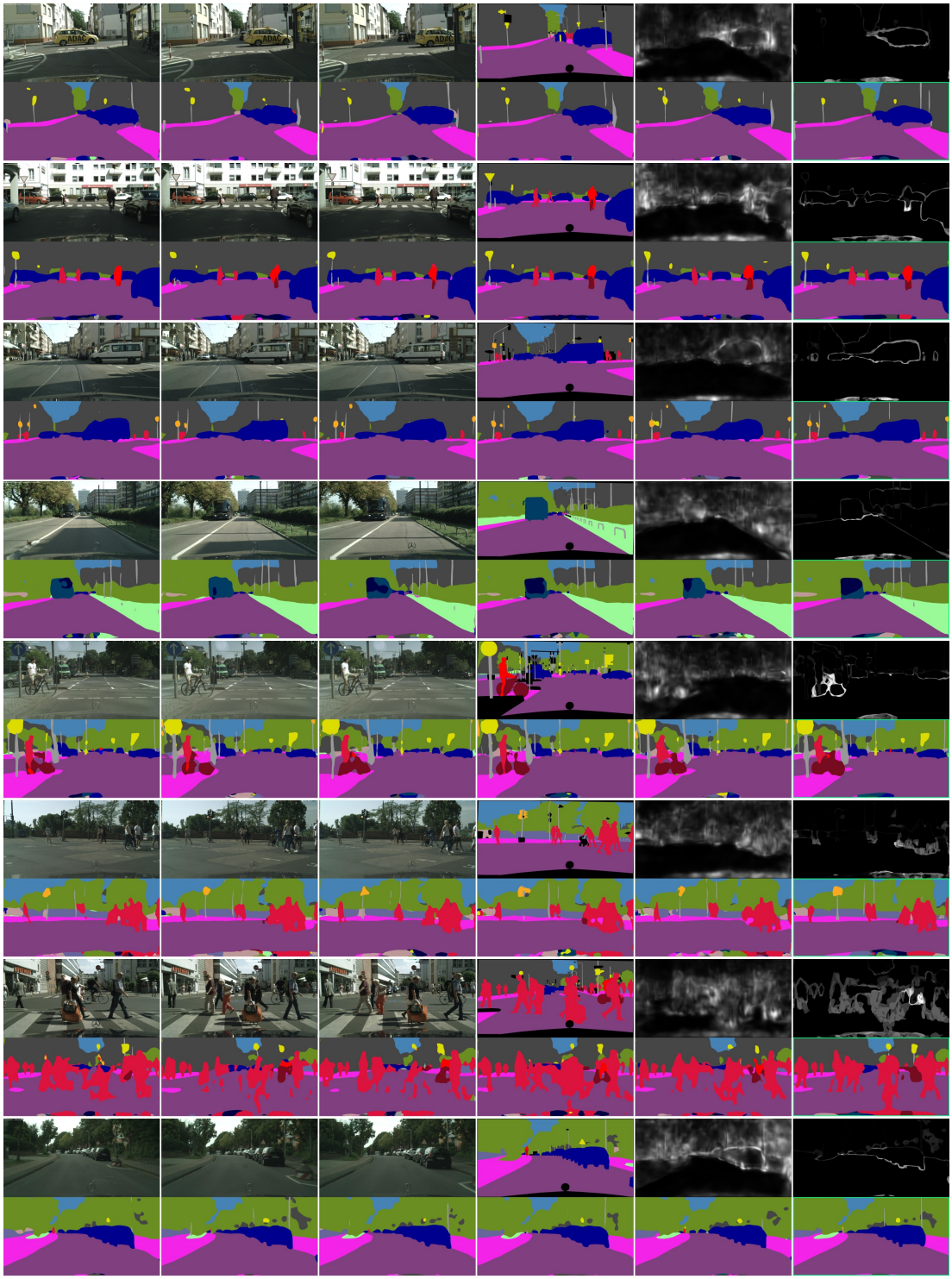
Metoda	Gubitak	mIoU	mIoU-MO	MSE	LPIPS
Oracle	L2	66.12	64.24	/	/
Segmentacija zadnje slike	L2	37.24	28.31	/	/
DeformF2F-8[26]	L2	$46.36^{\pm 0.44}$	$40.78^{\pm 0.99}$	/	/
MM-DeformF2F	MR1	$46.23^{\pm 0.28}$	$41.07^{\pm 0.57}$	$2.81^{\pm 0.32}$	0.1049
MM-DeformF2F usrednjen	MR1	$46.96^{\pm 0.21}$	$41.90^{\pm 0.53}$	/	/
MM-DeformF2F	MR2	$37.48^{\pm 0.31}$	$29.66^{\pm 0.60}$	$7.42^{\pm 0.44}$	0.2279
MM-DeformF2F usrednjen	MR2	$40.32^{\pm 0.12}$	$32.42^{\pm 0.37}$	/	/

Tablica 10.2: Rezultati srednjoročnog predviđanja. Uz korištenje gubitka MR1 naš model postiže nešto bolji mIoU, dok su uz gubitak MR2 predikcije puno raznovrsnije.

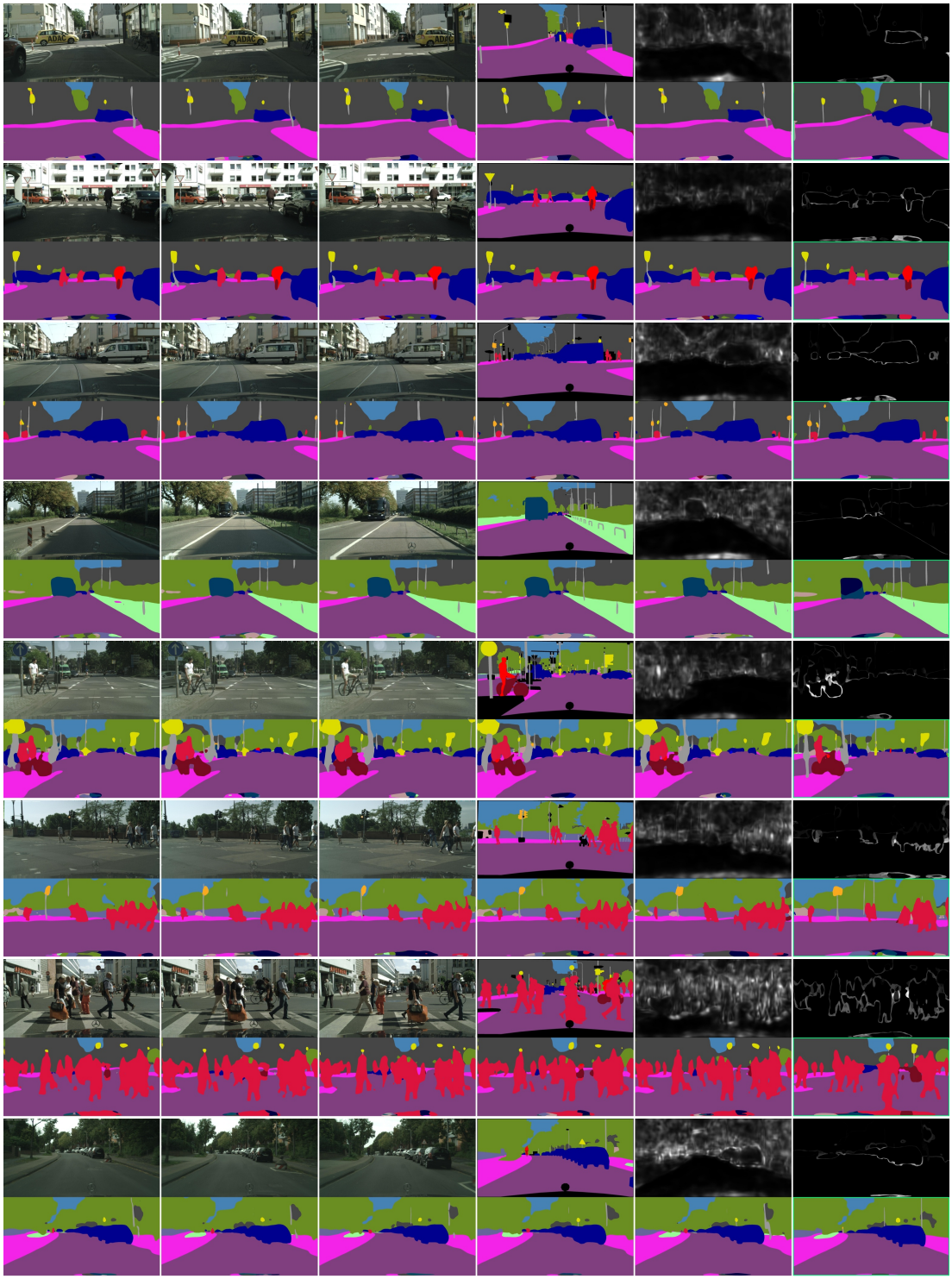
U nastavku su prikazane predikcije dobivene prilikom kratkoročnog i srednjoročnog predviđanja uz korištenje gubitaka MR1 i MR2. Zadnja predikcija u drugom redu svake scene je dobivena na osnovnom modelu [26] i služi za usporedbu.



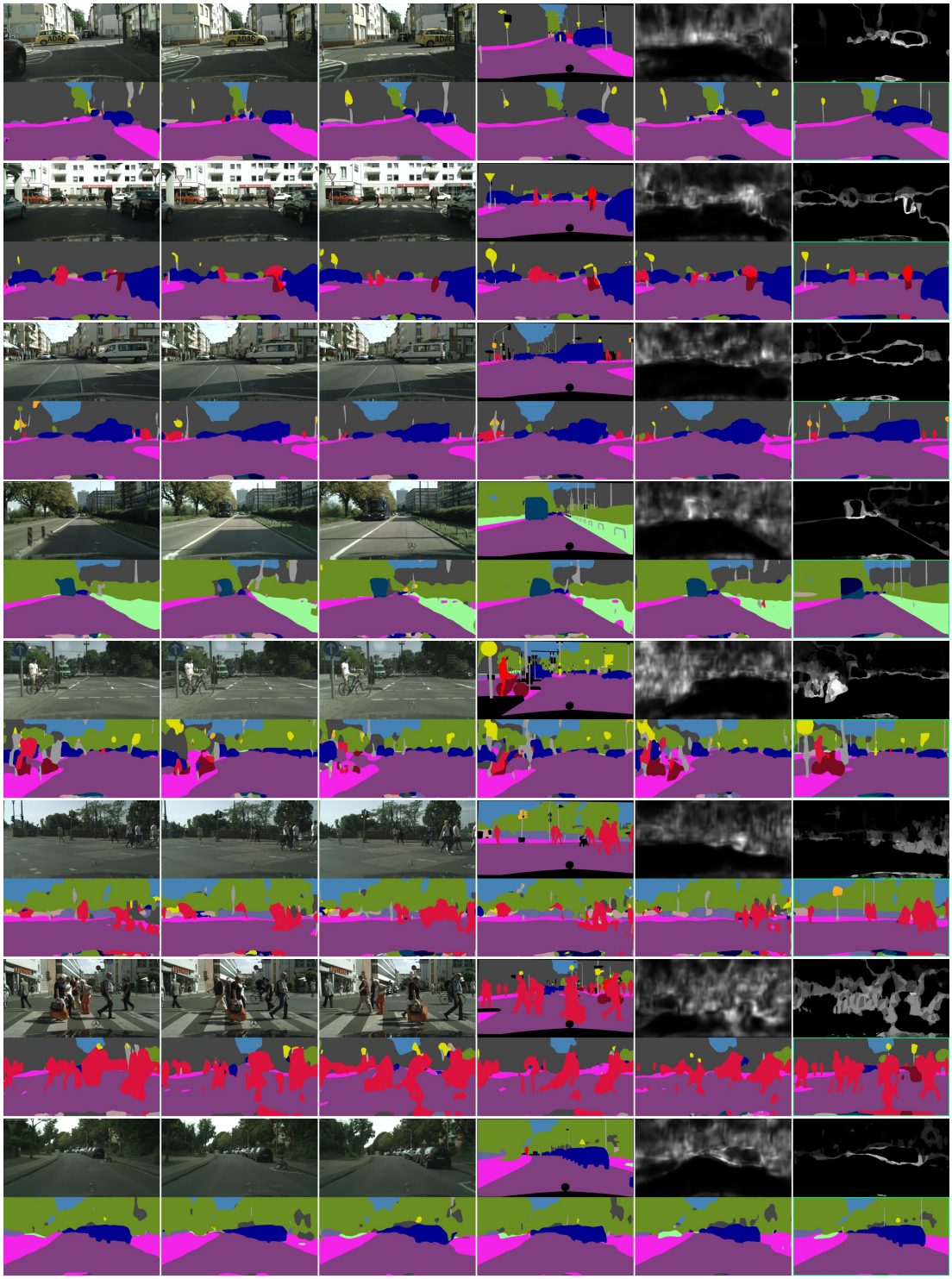
Slika 10.6: Predviđanje kratkoročne budućnosti s gubitkom MR1. Za svaku grupu slika u prvom redu su prikazani: prva i zadnja slika koje je model vidio (od ukupno četiri), slika iz budućnosti, njen *ground-truth*, te srednja varijanca logita po mapama značajki i varijanca diskretnih predikcija. U drugom redu prikazano je pet dobivenih predikcija, dok je na zadnjoj slici (zeleni okvir) prikazana predikcija dobivena modelom iz [26] s mIoU 58.98.



Slika 10.7: Predviđanje kratkoročne budućnosti s gubitkom MR2. Za svaku grupu slika u prvom redu su prikazani: prva i zadnja slika koje je model vidio (od ukupno četiri), slika iz budućnosti, njen *ground-truth*, te srednja varijanca logita po mapama značajki i varijanca diskretnih predikcija. U drugom redu prikazano je pet dobivenih predikcija, dok je na zadnjoj slici (zeleni okvir) prikazana predikcija dobivena modelom iz [26] s mIoU 58.98.



Slika 10.8: Predviđanje srednjoročne budućnosti s gubitkom MR1. Za svaku grupu slika u prvom redu su prikazani: prva i zadnja slika koje je model vidio (od ukupno četiri), slika iz budućnosti, njen *ground-truth*, te srednja varijanca logita po mapama značajki i varijanca diskretnih predikcija. U drugom redu prikazano je pet dobivenih predikcija, dok je na zadnjoj slici (zeleni okvir) prikazana predikcija dobivena modelom iz [26] s mIoU 58.98.



Slika 10.9: Predviđanje srednjoročne budućnosti s gubitkom MR2. Za svaku grupu slika u prvom redu su prikazani: prva i zadnja slika koje je model vidio (od ukupno četiri), slika iz budućnosti, njen *ground-truth*, te srednja varijanca logita po mapama značajki i varijanca diskretnih predikcija. U drugom redu prikazano je pet dobivenih predikcija, dok je na zadnjoj slici (zeleni okvir) prikazana predikcija dobivena modelom iz [26] s mIoU 58.98.

10.2.1. Utjecaj broja generiranih predikcija na performanse modela

Na tablici 10.3 prikazan je utjecaj broja generiranih predikcija (K) na mjeru mIoU i raznolikost semantičke segmentacije. Može se primijetiti da veći broj generiranih predikcija doprinosi raznolikosti i neznatno smanjuje mIoU. Prilikom učenja koristimo K=8 s kojim postižemo zadovoljavajuće performanse uz prihvatljivo vrijeme učenja. Učenje modela uz K=16 trajalo bi otprilike duplo duže.

K	mIoU	MSE	LPIPS
16	45.28	4.105	0.1496
8	45.81	3.004	0.1264
4	46.48	2.078	0.0947
2	46.44	1.424	0.0608
1	46.32	0.014	0.0010

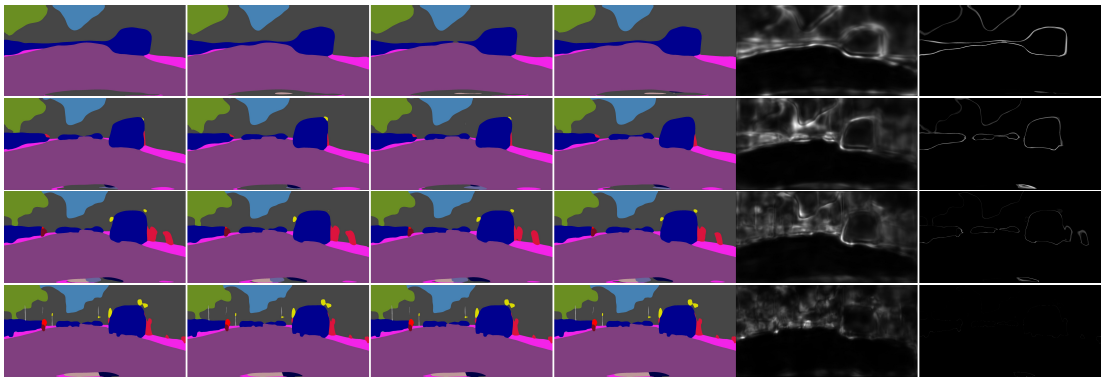
Tablica 10.3: Utjecaj broja generiranih uzoraka (K) na mIoU i raznolikost semantičke segmentacije kod srednjoročnog predviđanja uz gubitak MR1. Može se primijetiti da veći broj generiranih uzoraka doprinosi raznolikosti i neznatno smanjuje mIoU. Testiranje se vršilo u ranoj fazi izrade rada, te se rezultati donekle razlikuju od onih iz tablice 10.2. Budući da rezultate temeljimo na samo jednom naučenom modelu za svaki K, zbog varijance koja nastaje i prilikom učenja i prilikom evaluacije razlike u mIoU ne odražavaju nužno stvarnu situaciju. Štoviše, na nekim zadacima učenje sa K=12 ili K=16 ponekad daje veću preciznost. Broj generiranih uzoraka naveden u tablici odnosi se na fazu učenja, dok se mjerenje mIoU, MSE i LPIPS vršilo na 8 generiranih uzoraka u fazi iskorištavanja.

10.2.2. Doprinos gubitka GAN-a

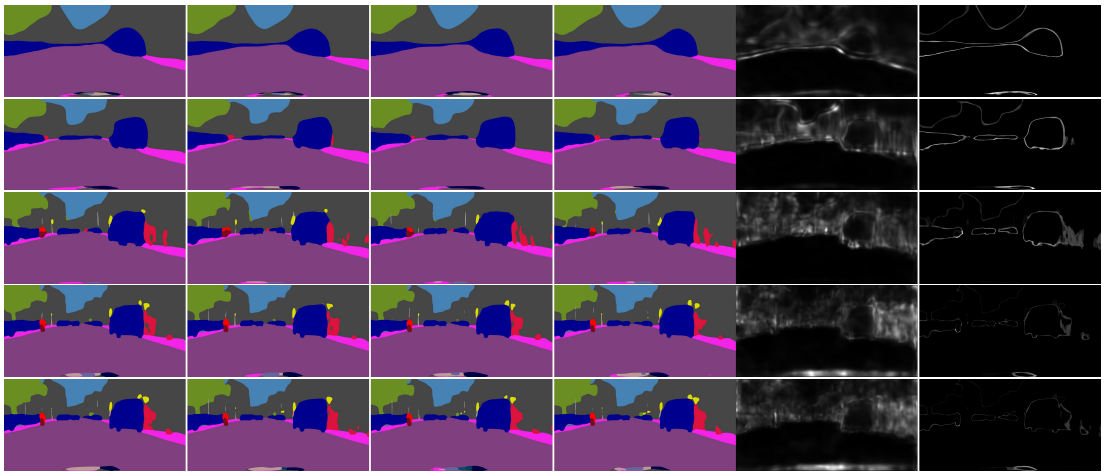
Kako bismo pokazali da raznolikost na izlazu nije samo posljedica korištenja gubitaka rekonstrukcije momenata i unošenja nasumičnih vrijednosti u model F2F, evaluirali smo model bez suparničkog gubitka ($\lambda_{GAN} = 0$). Predviđamo kratkoročnu budućnost uz gubitak MR1, a generator pri učenju proizvodi 8 predikcija za pojedini ulaz.

Iako su neke raznolikosti na početku vidljive (MSE oko 0.7), već oko 12-te epohe raznolikost se sve manje vidi (MSE oko 0.35), a poslije 40-te je nezamjetna (MSE oko 0.1). Model je najbolji mIoU koji iznosi 59.18 postigao na epohi 160 (iako je pušten

400 epoha), a mjera MSE na toj epohi iznosila je 0.08. Na slici 10.10a je vidljivo poboljšanje performansi kroz epohe, ali i postupno slabljenje raznolikosti. Odabrana je slika s puno neoznačenih (eng. *void*) površina zato što je obično na tim mjestima najveća raznolikost. Na slici 10.10b je ista ta scena, ali predikcije su dobivene s modelom učenim s težinama $\lambda_{GAN} = 10$ i $\lambda_{MR} = 100$. Modelu je trebalo 232 epohe da postigne svoj najbolji mIoU koji iznosi 59.14[1], ali MSE se držao stabilno iznad 1 sve do zadnje, 400-te, epohe.



(a) $\lambda_{GAN} = 0$



(b) $\lambda_{GAN} = 10$

Slika 10.10: Na prve četiri slike u svakom redu prikazane su generirane semantičke segmentacije, na predzadnjoj slici prikazana je srednja varijanca logita po mapama značajki, a na zadnjoj varijanca diskretnih predikcija. Primjerci su generirani u epohama 1, 5, 13 i 161 (i 232 na slici b). Prikazano je 4 od ukupno 8 generiranih uzoraka.

[1]Ovdje razmatramo samo rezultate dobivene tijekom učenja, odnosno evaluacijom koja se vrši svake 4 epohe, te je 59.14 najbolji mIoU zabilježen. Budući da se pri svakom unaprijednom prolazu unosi nasumičnost, evaluacija više puta na istom modelu neće nužno dati isti mIoU, stoga drugdje u radu brojke mogu biti drugačije.

10.2.3. Vrijeme učenja i zaključivanja

Učenje generativnih modela općenito traje duže od klasičnih, pa tako ovdje naš model učimo kroz 400[2] epoha, dok na neizmijenjenoj implementaciji model učimo kroz 160 epoha. Učenje s veličinom mini-grupe 8 i s 8 generiranih uzoraka na grafičkoj kartici GTX 970 traje u prosjeku 6 minuta po epohi, dok je na neizmijenjenoj implementaciji([26]) ta brojka šest puta manja. Učenje našeg modela na 400 epoha traje u prosjeku 40 sati, dok učenje osnovnog modela na 160 epoha traje u prosjeku nešto manje od 3 sata. U vrijeme je uračunata evaluacija svake 4 epohe (pri evaluaciji se generira samo jedna predikcija neovisno o postavljenom broju generiranih uzoraka generatora prilikom učenja) , ali ne i spremanje generiranih predikcija. Zaključivanje modela odvija se približno jednakom brzinom, pa je tako za vrednovanje 500 slika iz skupa za validaciju osnovnom modelu potrebno 2 minute i 7 sekundi, dok je našem modelu potrebno oko 5 sekundi više. U to vrijeme je uračunat kompletan prolaz slike kroz granu za izvlačenje značajki, F2F i granu za naduzorkovanje.

Tip predviđanja	Gubitak	Epoha s najboljim rezultatom (od 400)
Kratkoročno	MR1	210
Kratkoročno	MR2	350
Srednjoročno	MR1	129
Srednjoročno	MR2	365

Tablica 10.4: Epohe s najboljim rezultatom za različite gubitke i tipove predviđanja, možemo primijetiti da modelima s gubitkom MR2 treba više vremena za postizanje dobrog rezultata. Prikazana je srednja vrijednost dobivena na 3 naučena modela za pojedini zadatak.

[2]Iako se to u nekim slučajevima pokazalo nepotrebno dugo kao što vidimo na tablici 10.4.

11. Zaključak

U ovom radu smo predstavili model za semantičko predviđanje višemodalne budućnosti u scenama iz prometa. Pokazali smo da uvođenje generativnih metoda u model za predviđanje budućih značajki iz prošlih može doprinijeti rezultatima. Proveli smo eksperimente na zadacima kratkoročnog i srednjoročnog predviđanja uz približno jednake hiperparametre. U slučaju kratkoročnog predviđanja dobivena je veća mIoU ocjena, uz primjetnu ali slabu raznolikost. Zadatak srednjoročnog predviđanja je nešto teži, pa je i nesigurnost modela te raznolikost primjeraka veća, što rezultate čini vizualno zanimljivijima, ali točnošću inferiornijima. Iako bi uz minimalne intervencije i kod srednjoročnog predviđanja mogli dobiti mIoU veći od osnovnog modela, to ipak ne radimo jer mIoU nije jedina relevantna mjera u ovakvom zadatku. Shodno tomu, iako korištenje gubitka MR2 smanjuje mjeru mIoU za 5 ili više postotnih poena, svejedno nam ga je zanimljivo koristiti zbog veće raznolikosti. Vizualno najzanimljivije predikcije dobili smo upravo uz korištenje gubitka MR2 na zadatku kratkoročnog predviđanja. Interesantno je vidjeti na generiranim slikama kada, gdje i s kojim razredima je model nesiguran i voljan riskirati. Pokazalo se da učenje novog modela traje 6 puta duže, ali vrijeme zaključivanja približno je osnovnom modelu što je bitno za zadatke poput autonomne vožnje. U budućim eksperimentima valjalo bi isprobati i gubitke *proxy* MR1 i *proxy* MR2, naučiti model na zadatku segmentacije instanci umjesto semantičke segmentacije, izmjeriti performanse na slikama pune rezolucije te pronaći način kako poboljšati generativni dio modela - primjerice novim načinom generiranja šuma, drugačijom arhitekturom diskriminatora, različitim optimizatorima i drugim metodama.

LITERATURA

[1] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, i Bernt Schiele. The cityscapes dataset for semantic urban scene understanding, 2016. URL https://arxiv.org/abs/1604.01685.

[2] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, i Yichen Wei. Deformable convolutional networks, 2017. URL https://arxiv.org/abs/1703.06211.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, i Li Fei-Fei. *ImageNet: A large-scale hierarchical image databas*, 2009. URL https://ieeexplore.ieee.org/document/5206848.

[4] Vincent Dumoulin i Francesco Visin. *A guide to convolution arithmetic for deep learning*, 2018. URL https://arxiv.org/pdf/1603.07285.pdf.

[5] Ross Girshick. Fast r-cnn, 2015. URL https://arxiv.org/abs/1504.08083.

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, i Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, 2013. URL https://arxiv.org/abs/1311.2524.

[7] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, i Yoshua Bengio. *Generative Adversarial Networks*, 2014. URL https://arxiv.org/abs/1406.2661.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, i Jian Sun. *Deep Residual Learning for Image Recognition*, 2015. URL https://arxiv.org/abs/1512.03385.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, i Ross Girshick. Mask r-cnn, 2017. URL https://arxiv.org/abs/1703.06870.

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, i Alexei A. Efros. *Image-to-Image Translation with Conditional Adversarial Networks*, 2016. URL https://arxiv.org/abs/1611.07004.

[11] Ivan Krešo, Siniša Šegvić, i Josip Krapac. Ladder-style DenseNets for semantic segmentation of large natural images. U *Proceedings of the IEEE International Conference on Computer Vision Workshops*, stranice 238–245, 2017.

[12] Ivan Krešo, Josip Krapac, i Siniša Šegvić. Efficient ladder-style DenseNets for semantic segmentation of large images. *IEEE Transactions on Intelligent Transportatoin Systems*, 2020.

[13] Alex Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[14] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, i Wenzhe Shi. *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*, 2016. URL https://arxiv.org/abs/1609.04802.

[15] Chuan Li i Michael Wand. *Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks*, 2016. URL https://arxiv.org/abs/1604.04382.

[16] William Lotter, Gabriel Kreiman, i David Cox. *Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning*, 2016. URL https://arxiv.org/abs/1605.08104.

[17] Pauline Luc, Camille Couprie, Yann LeCun, i Jakob Verbeek. Predicting future instance segmentation by forecasting convolutional features, 2018. URL https://arxiv.org/abs/1803.11496.

[18] Luke Metz, Ben Poole, David Pfau, i Jascha Sohl-Dickstein. *Unrolled generative adversarial networks*, 2016. URL https://arxiv.org/abs/1611.02163.

[19] Mehdi Mirza i Simon Osindero. *Conditional Generative Adversarial Nets*, 2014. URL https://arxiv.org/abs/1411.1784.

[20] Marin Oršić, Ivan Krešo, Petra Bevandić, i Siniša Šegvić. In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images, GCPR 2019. URL https://arxiv.org/abs/1903.08469.

[21] Shaoqing Ren, Kaiming He, Ross Girshick, i Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2015. URL https://arxiv.org/abs/1506.01497.

[22] Lee Soochan, Ha Junsoo, i Kim Gunhee. *Harmonizing Maximum Likelihood with GANs for Multimodal Conditional Generation*, 2019. URL https://soochanlee.com/publications/mr-gan.

[23] Zhou Wang, A.C. Bovik, H.R. Sheikh, i E.P. Simoncelli. *Image quality assessment: from error visibility to structural similarity*, 2004. URL https://www.cns.nyu.edu/pub/eero/wang03-reprint.pdf.

[24] Richard Zhang, Phillip Isola, Alexei A. Efros, i Eli Shechtman andOliver Wang. *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*, 2018. URL https://richzhang.github.io/PerceptualSimilarity/.

[25] Josip Šarić, Marin Oršić, Tonći Atunović, Sacha Vražić, i Siniša Šegvić. Warp to the future: Joint forecasting of features and feature motion. U *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[26] Josip Šarić, Marin Oršić, Tonći Antunović, Sacha Vražić, i Siniša Šegvić. *Single Level Feature-to-Feature Forecasting with Deformable Convolutions*, CVPR 2019. URL https://arxiv.org/pdf/1907.11475.pdf.

[27] Siniša Šegvić. *Konvolucijski modeli, prezentacija s predmeta Duboko učenje na Fakultetu elektrotehnike i računarstva*. URL http://www.fer.unizg.hr/_download/repository/UI_12_UmjetneNeuronskeMreze[1].pdf.

Predviđanje višemodalne semantičke budućnosti u videu cestovne scene

Sažetak

U ovom radu razmatramo semantičko predviđanje višemodalne budućnosti. Glavni problem u tom kontekstu predstavljaju novootkriveni prostori i artikulirani objekti kod kojih se i u kratkoročnom slučaju suočavamo s velikom nesigurnošću predviđanja. Ideja ovog rada je dopustiti modelu nešto više slobode u smislu mogućnosti predviđanja više različitih budućnosti. To činimo pretvaranjem osnovnog regresijskog modela u uvjetni generativni model temeljen na suparničkom učenju primjenom gubitaka rekonstrukcije momenata. U slučaju kratkoročnog predviđanja dobivena je veća točnost modela, dok kod srednjoročnog predviđanja imamo nešto nižu točnost, ali rezultati su raznoliki i vizualno zanimljiviji. Pokazalo se da učenje novog modela traje 6 puta duže, a vrijeme zaključivanja pribiližno je osnovnom modelu.

Ključne riječi: duboko učenje, računalni vid, generativni suparnički modeli, GAN, multimodalno generiranje slika, MR-GAN, predviđanje semantičke budućnosti, predviđanje iz značajki u značajke, F2F, semantička segmentacija, segmentacija instanci.

Multimodal semantic forecasting in video

Abstract

In this paper we address the multimodal semantic forecasting in road driving scenarios. The main problem in this context is great uncertainty in forecasting of articulated objects and unoccluded spaces, even in the case of short-term predictions. The idea of this paper is to give the model somewhat more freedom by allowing it to predict multiple futures. We do this by converting the base regression model into a conditional generative adversarial model and by using moment reconstruction losses. In the case of short-term prediction, higher accuracy of the model was obtained, while in the case of long-term prediction, we observe slightly lower accuracy, but the predictions are diverse and visually more interesting. It takes about 6 times longer to train the model, while the evaluation time is similar to the base model.

Keywords: deep learning, computer vision, generative adversarial models, GAN, multimodal image generation, MR-GAN, semantic future prediction, feature-to-feature, F2F, semantic segmentation, instance segmentation.