SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 203

VALIDIRANJE HIPERPARAMETARA ZA SEMANTIČKO PROGNOZIRANJE U VIDEU

Jakov Rukavina

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 203

VALIDIRANJE HIPERPARAMETARA ZA SEMANTIČKO PROGNOZIRANJE U VIDEU

Jakov Rukavina

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 10. ožujka 2023.

DIPLOMSKI ZADATAK br. 203

Pristupnik:	**Jakov Rukavina (0036509435)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	prof. dr. sc. Siniša Šegvić

Zadatak: **Validiranje hiperparametara za semantičko prognoziranje u videu**

Opis zadatka:

Prognoziranje budućnosti u videu neriješen je problem računalnog vida s mnogim zanimljivim primjenama. Posebno su zanimljiva rješenja utemeljena na semantičkoj informaciji koju pružaju duboki modeli za gustu predikciju. Takav pristup omogućava atraktivne primjene zasnovane na izravnom prognoziranju semantičkog sadržaja u videu. U okviru rada, potrebno je proučiti duboke arhitekture za semantičko predviđanje cestovnih scena u videu. Obratiti pažnju na pristupe utemeljene na regresiranju pomaka značajki. Oblikovati arhitekturu te pobrojati stupnjeve slobode. Uhodati učenje modela te validiranje hiperparametara. Prikazati i ocijeniti postignutu generalizacijsku izvedbu. Predložiti pravce budućeg razvoja. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, kao i potrebna objašnjenja te dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 23. lipnja 2023.

Zahvaljujem se prof. dr. sc. Siniši Šegviću i dr. sc. Josipu Šariću na prenesenom znanju, savjetima i strpljenju tijekom pisanja ovog rada. Također se zahvaljujem roditeljima, Ledi i prijateljima na najboljoj podršci tijekom cijelog studija.

Sadržaj

Uvod

Područje dubokog učenja iznimno je popularno posljednjih godina, a posebno je mnogo napretka postignuto primjenom dubokih modela u području računalnog vida. Duboki modeli omogućili su razne nove primjene, između ostaloga potaknuli su i razvoj autonomnih vozila, koja se najčešće za percepciju oslanjaju upravo na slike dobivene kamerama postavljenim na vozilo. U sustavima za autonomnu vožnju vrlo je bitno omogućiti pravovremeno i proračunato donošenje odluka. Pri većim brzinama ostaje vrlo malo vremena za donošenje takvih odluka što može životno ugroziti putnike i ostale sudionike prometa ili u najboljem slučaju izazvati materijalnu štetu. Mogući pristup ovom problemu je prognoziranje budućih stanja u prometu kako bismo mogli predvidjeti potencijalnu opasnost koja se još nije realizirala. Jedan od načina na koji to pokušavamo postići jest prognoziranjem budućih semantičkih oznaka.

U ovom radu razmatramo prognoziranje značajki dobivenih iz segmentacijskog modela opisanog prvim poglavljem. Prognoziranjem značajki postižemo kompetitivne rezultate uz mnogo manju računsku složenost u odnosu na druge pristupe semantičkom prognoziranju. Implementirat ćemo nekoliko arhitektura za prognoziranje semantičkih značajki iz literature. Nadalje, eksperimentalno ćemo isprobati više predloženih modifikacija implementiranih arhitektura, opisanih u drugom poglavlju, s ciljem poboljšanja točnosti. Pritom ćemo posebno paziti na utjecaj modifikacija na povećanje računske složenosti. Sve modifikacije ćemo detaljno opisati te evaluirati u petom poglavlju s eksperimentalnim rezultatima.

Osim semantičkog prognoziranja, razmatramo i uporabu istih arhitektura za prognoziranje latentnih reprezentacija iz generativnog modela opisanog u trećem poglavlju. Prognozirane latentne reprezentacije rekonstruiramo dekoderom generativnog modela čime dobivamo predviđenu sliku. Ovakav pristup rijetko se susreće u literaturi, a mi ga koristimo za prognoziranje budućih RGB slika u videu. U petom poglavlju predlažemo i multimodalnu arhitekturu za združeno prognoziranje semantičkih i generativnih značajki. Motivacija za združeno prognoziranje dolazi od pretpostavke kako razmatrani modeli mogu postići bolje rezultate prognozirajući semantičke značajke. Stoga za modeliranje budućih reprezentacija generativnog modela nastojimo iskoristiti izlučene informacije o pomacima koje dobivamo temeljem semantičkih značajki. Konačno, sve eksperimente provest ćemo na skupu Cityscapes, koji ćemo predstaviti u četvrtom poglavlju.

1. Duboki modeli za gusto raspoznavanje

Gusto raspoznavanje iz slike podrazumijeva zadatke u kojima svakom pikselu slike treba pridijeliti oznaku. To uključuje zadatke poput semantičke segmentacije, procjene dubine te detekcije objekata. Pritom se često koriste duboki modeli namijenjeni rješavanju više zadataka dijeljenjem značajki koje se u posljednjem koraku prerađuju specijaliziranim glavama za određen zadatak (eng. *multi-head learning*). Dijeljenje značajki u usporedbi s klasičnim modelom specijaliziranim za jedan zadatak ima više prednosti. Prvenstveno, dijeljenjem parametara modela smanjujemo memorijsko zauzeće te računsku složenost budući da nije potrebno više puta izračunavati značajke za svaki zadatak. Također, dijeljenje značajki može imati regularizacijski učinak te pospješiti performanse modela ako zadaci dijele komplementarne informacije [6]. Klasifikacijski modeli predtrenirani na velikim skupovima podataka poput ImageNeta ključni su dio modernih arhitektura za gusto raspoznavanje. Koriste se kao ekstraktor značajki koji primjenjujemo direktno na sliku te ih obično nazivamo okosnicom modela (eng. *backbone*) [2]. U ovom poglavlju opisat ćemo korištenu konvolucijsku arhitekturu za semantičku segmentaciju te uobičajene prakse korištene kod potpuno konvolucijskih modela.

1.1. Okosnica ResNet

Odabrana arhitektura za semantičku segmentaciju koristi predtrenirani ResNet kao okosnicu. Predtreniranje okosnice na skupu slika ImageNet omogućuje prijenos znanja stečenog učenjem na jeftinim, te brojnim, klasifikacijskim oznakama na razini cijele slike [2]. Učenjem potpunog segmentacijskog modela parametri okosnice imaju priliku fino se prilagoditi specifičnostima zadatka semantičke segmentacije, ili pak potpuno prenamijeniti dio težina, ako je to potrebno. Ovakav pristup omogućuje brže treniranje, poboljšava generalizaciju te postiže bolje performanse modela. Navedene prednosti proizlaze iz činjenice da skupovi s gustim oznakama u pravilu sadrže manji broj slika budući da je njihovo označavanje značajno skuplje. Posljedično, predtrenirana okosnica „vidjela“ je mnogo više slika pa često može prepoznati vizualne koncepte koji su prisutni i u ciljanom skupu podataka [2]. Možda je još važnije primijetiti kako će model iz slučajne inicijalizacije vrlo teško naučiti prepoznati određene aspekte koji su slabo zastupljeni u ciljanom skupu.

Skupovi slika poput ImageNeta sadrže veliku raznolikost slika što pogoduje učenju slabije zastupljenih primjera.

Arhitektura ResNet, predstavljena 2015. godine, potaknula je revoluciju u svijetu dubokog učenja pokazavši da je moguće naučiti još dublje modele, te je pritom nadmašila performanse dotadašnjih konvolucijskih arhitektura. Glavni novitet ResNeta je uvođenje rezidualnih veza, od kojih potječe i naziv same arhitekture. Rezidualna veza je preskočna veza koja izlazu parametriziranih slojeva $F(x)$ nadodaje identitet x. Budući da su $F(x)$ i x tenzori istih dimenzija, operator $+$ predstavlja zbrajanje po elementima, a konačni rezultat možemo zapisati kao $F(x) + x$. Navedenu operaciju provodit će skup slojeva prikazan na slici 1.1, koji obično nazivamo rezidualni blok (eng. *residual block*). Da ne bi došlo do zabune, rezidualni blok također uključuje i aktivacijsku funkciju koju označavamo sa σ, što možemo preciznije zapisati izrazom 1.

$$resblock := \sigma(F(x) + x) \tag{1}$$

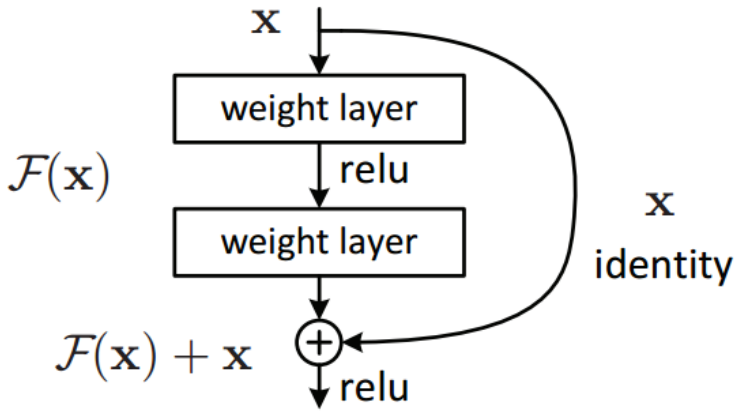
Rezidualne veze olakšavaju posao težinskim slojevima u rezidualnom bloku. Težinski slojevi više ne moraju direktno naučiti željeno preslikavanje $H(x)$, već je dovoljno naučiti rezidualnu funkciju iz izraza 2. Autori ResNeta u radu [5] argumentiraju kako je lakše optimizirati rezidualno preslikavanje nego zamišljeno željeno preslikavanje. Dodatno, kada bi željeno preslikavanje bilo jednako identitetu, bilo bi lakše pritegnuti težinsku transformaciju na nulu nego postići funkciju identiteta nelinearnim slojevima. U realnim slučajevima malo je vjerojatno da je željeno preslikavanje jednako identitetu. Međutim, ako je željena funkcija bliža identitetu nego preslikavanju u nulu, optimizatoru će biti lakše pronaći rezidual uz preskočnu vezu nego naučiti potpuno novo preslikavanje.

$$F(x) := H(x) - x \tag{2}$$

Cilj dodavanja rezidualnih veza zapravo je bio omogućiti učenje puno dubljih modela nego što je to prije bilo moguće. Eksperimenti iz rada [5] pokazali su da dodavanjem konvolucijskih slojeva točnost prvo stagnira, a onda počinje i padati. Ovo opažanje nije uzrokovano prenaučenošću modela (eng. *overfitting*) jer dodavanje slojeva ne rezultira samo padom točnosti na skupu za testiranje, već i na skupu za treniranje. To je kontraintuitivno jer bi dublji model u najmanju ruku trebao moći „oponašati“ model s manje slojeva tako da u nadodanim slojevima nauči preslikavati identitet. Autori zaključuju kako tadašnji

optimizatori nisu mogli pronaći rješenja koja su podjednako dobra kao kod jednostavnijih modela ili bar to nisu uspjeli postići u razumnom vremenu.

Također, tu je i problem nestajućih gradijenata (eng. *vanishing gradients*) koji se pojavljuje jer dodavanjem slojeva modelu dodajemo i aktivacijske funkcije. Propagacijom pogreške unatrag kroz aktivacije poput sigmoide gradijenti se eksponencijalno smanjuju što se više približavamo početnim slojevima. Ovaj problem ublažen je korištenjem aktivacijske funkcije ReLU (eng. *rectifed linear unit*) i normalizacije po grupi (eng. *batch normalization*), a dodavanje rezidualnih veza dodatno olakšava učenje prvih slojeva jer gradijenti sada do njih dolaze gotovo izravno. Eksperimenti su pokazali da uz rezidualne veze pogreška opada i kod puno dubljih modela. Rad [5] utvrdio je da je sada moguće učiti rezidualne arhitekture koje se sastoje od više stotina slojeva.



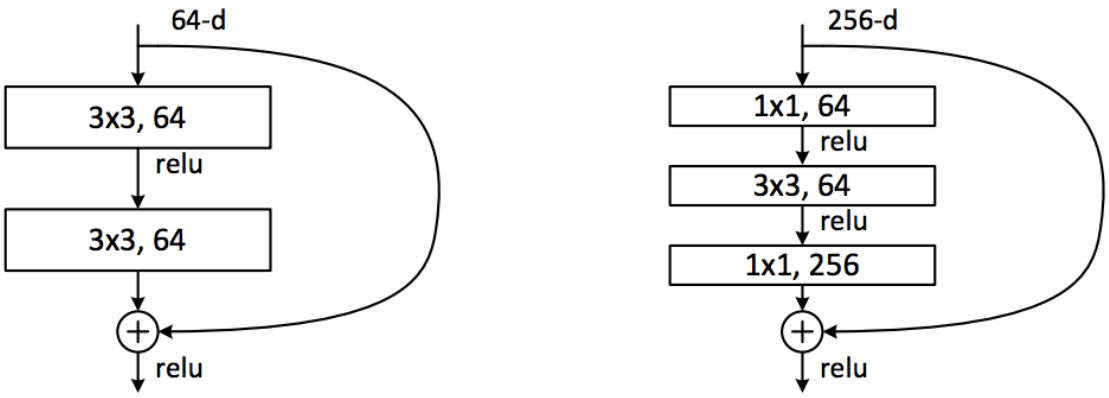
Slika 1.1: Rezidualni blok korišten u arhitekturi ResNet. Izlazu parametriziranih slojeva $F(x)$ nadodaje se identitet x preskočnom (rezidualnom) vezom. Rezultat se dobiva zbrajanjem po elementima te aktivacijom ReLU. Slika preuzeta iz [5].

Također, treba napomenuti kako ResNet implementira i neke druge važne koncepte osim rezidualnih veza. Autori uvode normalizaciju po grupi (eng. *batch normalization*) iz rada [8] postavljajući normalizirajući sloj između konvolucijskih slojeva i aktivacijske funkcije. Uvođenje normalizacije između težinskih slojeva zaglađuje funkciju gubitka [2] što zauzvrat znatno ubrzava treniranje modela te omogućuje postizanje bolje točnosti.

Autori ResNeta su kasnije u radu [7] napravili promjene u poretku slojeva rezidualnog bloka. Eksperimentalno je pokazano kako model postiže bolje performanse ako normalizacija po grupi prethodi aktivacijskoj funkciji i konvolucijskom sloju. Još jedan važan iskorak je uvođenje globalnog sažimanja (eng. *global pooling*) na koji se nadovezuje jedan potpuno

povezani sloj. Ova izmjena odnosi se na klasifikacijski ResNet, no bitna je jer se zamjenom višestrukih potpuno povezanih slojeva jednim težinskim slojem smanjuje broj parametara modela te prebacuje težište kapaciteta na konvolucijske slojeve [2]. Trend prebacivanja težišta s potpuno povezanih slojeva na konvolucije usadio se u moderne arhitekture neuronskih mreža.

U radu [5] predstavljene su dvije verzije rezidualnog bloka, a nazvat ćemo ih „osnovni“ rezidualni blok i rezidualni blok „s uskim grlom“. Osnovni rezidualni blok prikazan je lijevo na slici 1.2, a sastoji se od dva kovolucijska sloja s jezgrom veličine 3×3 gdje tijekom prolaska kroz cijeli blok broj kanala ostaje konstantan. Na desnoj strani slike 1.2 prikazan je rezidualni blok s uskim grlom. On se, pak, sastoji od 1×1 konvolucije koja smanjuje broj kanala, 3×3 konvolucije u sredini, te 1×1 konvolucije koja vraća broj kanala na početnu vrijednost. Motivacija iza uvođenja uskog grla (smanjivanja broja kanala) bila je smanjiti računsku složenost te broj parametara kod dubljih modela. Zato ResNet arhitekture s 50 i više slojeva koriste rezidualni blok s uskim grlom, dok modeli poput ResNeta-18 i ResNeta-34 koriste osnovni blok.



Slika 1.2: Osnovni rezidualni blok (lijevo) i rezidualni blok s uskim grlom (desno). Slika preuzeta iz [5].

ResNet-18 najmanji je model predstavljen radom [5], no on svejedno pruža kompetitivne performanse s obzirom na računsku složenost. Značajke ResNeta-18 nakon zadnjeg konvolucijskog sloja imaju 512 kanala te su na 32 puta manjoj rezoluciji od ulazne kao i kod drugih varijanti arhitekture slike (u daljnjem tekstu /32).

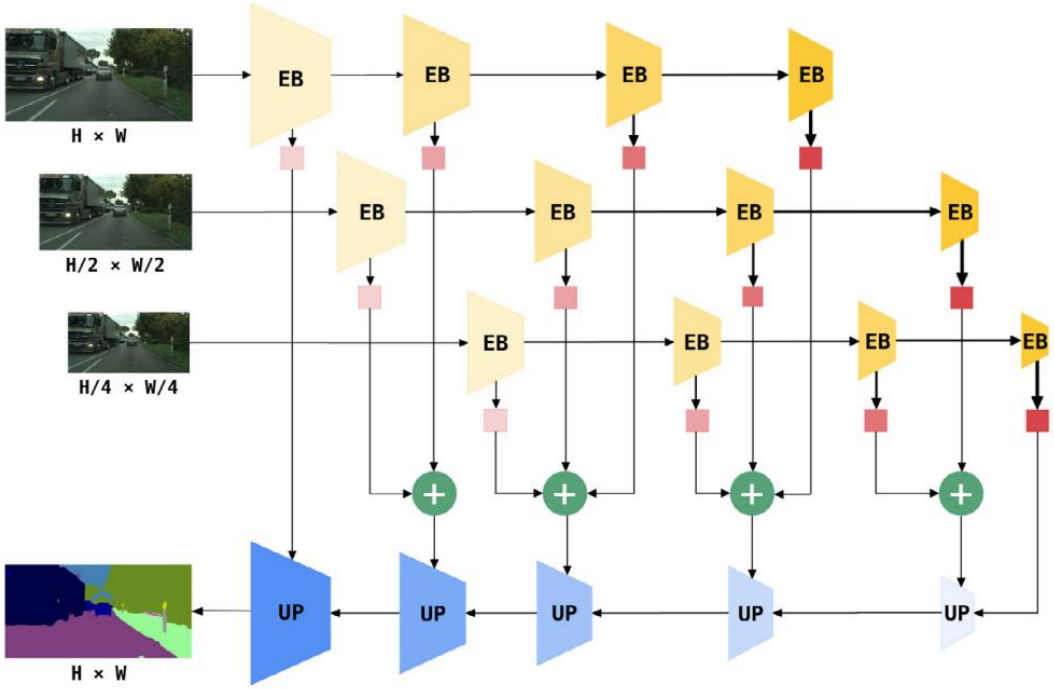
1.2. Piramidalni SwiftNet

Arhitektura SwiftNet osmišljena je primarno za semantičku segmentaciju, no može se primijeniti i na druge zadatke guste predikcije. Odlikuje ju mogućnost zaključivanja u stvarnom vremenu uz kompetitivne performanse. SwiftNet to postiže koristeći relativno male okosnice poput ResNeta-18 ili MobileNeta-v2 [4][3]koji imaju javno dostupne parametre modela treniranih na ImageNetu. Arhitekturu karakterizira asimetričan enkoder-dekoder jer autori smatraju da zadatak prepoznavanja značajki kojeg odrađuje enkoder zahtijeva puno više kapaciteta nego raspoznavanje granica klasa (uz već poznati grubi semantički kontekst) i naduzorkovanje, što su zadaće dekodera [3]. Postoje dvije, značajno različite, inačice arhitekture: jednorazinski i piramidalni SwiftNet. Iako bi se puno toga moglo reći i o jednorazinskom modelu, fokus ovog rada bit će na piramidalnom SwiftNetu koji koristi okosnicu ResNet-18.

Odabrana inačica arhitekture koristi rezolucijsku piramidu za dobivanje značajki iste slike. Enkoder se primjenjuje na sve slike iz rezolucijske piramide uz dijeljenje parametara po rezolucijskim razinama što pomaže regularizaciji. Time se omogućuje prepoznavanje objekata različitih veličina (često zbog različite udaljenosti u odnosu na kameru) bez povećavanja broja parametara enkodera. Primjenjivanjem enkodera na slike niske rezolucije postižemo veliko receptivno polje (u odnosu na scenu), dok je sa slikom originalne rezolucije receptivno polje manje, ali su značajke finije. Svaka razina piramide poduzorkuje sliku na dvostruko manju rezoluciju u odnosu na prethodnu sliku. Posljedično, gornja granica dodatne računske složenosti zbog uzastopnog korištenja enkodera na različitim razinama iznosi ~33% [3]. Rad [4] pokazao je kako model s tri razine piramide postiže najbolje performanse i takva je konfiguracija prikazana na slici 1.3, a mi ćemo još koristiti i konfiguraciju s dvije razine.

Okosnica (enkoder) modela podijeljena je na četiri stadija u kojima je rezolucija konstantna. Tijekom prvog stadija rezolucija značajki smanjuje se na /4, a nakon svakog sljedećeg stadija rezolucija se dodatno smanjuje za faktor 2×. Stoga su rezolucije u stadijima /4, /8, /16 i /32 u odnosu na rezoluciju ulazne slike. Enkoder je na slici 1.3 označen žutom bojom, a ista nijansa boje označava dijeljenje značajki istog stadija. Dekoder podsjeća na oblik ljestava jer svaki stadij prima značajke niže rezolucije iz prošlog stadija, te preskočne (lateralne) značajke više rezolucije iz enkodera. Za razliku od enkodera, značajke dekodera u svim stadijima imaju konstantan broj kanala. Zato je potrebno prilagoditi broj kanala u

preskočnim vezama kako bi oni odgovarali broju mapi značajki iz dekodera. To se postiže 1×1 konvolucijama, na slici 1.3 prikazanim crvenima kvadratima. Nakon projekcije u željen broj kanala, preskočne značajke iste rezolucije iz različitih slojeva piramide zbrajaju se po elementima te dalje prosljeđuju odgovarajućem stadiju za naduzorkovanje. Ovaj pristup naziva se piramidalna fuzija [3] te odstupa od rada [4] u kojem su mape značajki konkatenirane nakon čega se koristila projekcijska konvolucija.

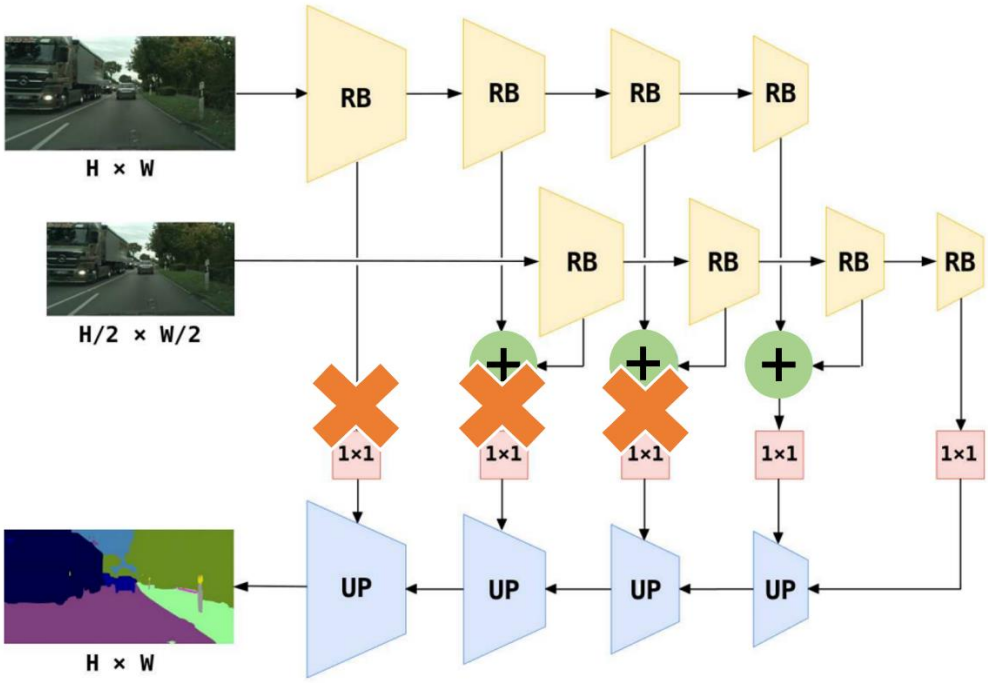


Slika 1.3: Trorazinski piramidalni SwiftNet. Žuti trapezi predstavljaju stadije enkodera koji se triput primjenjuje na slike: originalne, $H/2 \times W/2$ i $H/4 \times W/4$ rezolucije. Crveni kvadrati označavaju projekcijske konvolucije koje podešavaju broj mapi značajki kako bi bile prikladne za dekoder. Zeleni plusevi predstavljaju zbrajanje preskočnih veza iste rezolucije po elementima. Plavi trapezi su stadiji dekodera koji kombiniraju značajke niže rezolucije iz prijašnjeg stadija te više (ciljne) rezolucije iz preskočnih veza. Slika preuzeta iz [3].

Stadiji dekodera su na slici 1.3 označeni plavom bojom. Svaki stadij koristi se za naduzorkovanje značajki na rezoluciju veću za faktor $2\times$ u odnosu na prijašnji stadij. Broj samih stadija i početna rezolucija značajki ovise o zadanom broju slojeva piramide. Naduzorkovanje sadrži tri koraka: (i) značajke niže rezolucije iz prijašnjeg stadija bilinearno se naduzorkuju (na rezoluciju preskočne veze), (ii) naduzorkovane reprezentacije zbroje se po elementima sa prekočnom vezom te (iii) rezultat zbrajanja miješa se jednom 3×3 konvolucijom [3]. Posljednji korak je naduzorkovati logite na ciljnu rezoluciju

originalne slike, jer zadnji stadij dekodera proizvodi značajke na /4 rezoluciji. Ovaj korak uvelike smanjuje računsku složenost budući da ne koristi konvolucijske slojeve na najvišoj rezoluciji značajki.

Ovaj rad razmatra modificiranu arhitekturu SwiftNet u kojoj je aktivno samo nekoliko početnih preskočnih veza nižih rezolucija. Preskočne veze ukidamo jer uslijed dekodera želimo ugraditi prognostički modul koji semantičke značajke iz prošlosti koristi za prognoziranje budućih značajki. Nakon što dobijemo prognozirane značajke ne želimo ih miješati s preskočnim vezama iz prošlih trenutaka pa te veze ne koristimo. Postoji više razloga za prognoziranje značajki niže rezolucije koje u piramidalnom SwiftNetu možemo dobiti iz prvih nekoliko stadija dekodera. Slika 1.4 prikazuje dvorazinski piramidalni SwiftNet zbog njegove jednostavnosti u odnosu na model s tri razine. Ovaj model koristi samo jednu preskočnu vezu koja do prvog stadija dekodera prosljeđuje značajke enkodera rezolucije /32. Narančasti križići označavaju veze koje su izbačene iz arhitekture, a to su ovdje posljednje tri preskočne veze. Naš pristup prognoziranju u ovakvoj bi konfiguraciji ubacio prognostički modul između prvog i drugog stadija dekodera. Primijetite kako ovdje preskočne veze odstupaju od slike 1.3 gdje je prikazana unaprijeđena inačica arhitekture iz [3]. U poglavlju 5.1 razmotrit ćemo utjecaj micanja preskočnih veza na performanse modela.



Slika 1.4: Dvorazinski piramidalni SwiftNet s jednom preskočnom vezom. Narančasti križići označavaju preskočne veze koje su izbačene iz arhitekture, ostale oznake su kao u slici 1.3. Slika prenamijenjena iz [4].

2. Semantičko prognoziranje

Sposobnost prognoziranja budućnosti omogućava donošenje pravovremenih i proračunatih odluka u inteligentnim sustavima. Prognoziranje je iznimno bitno kod primjena poput autonomne vožnje [1] i raznih robotskih zadataka, u kojima jedna kriva odluka može ugroziti ljudski život.

U ovom radu fokus će biti na autonomnoj vožnji te prognoziranju budućih semantičkih oznaka, što će se odraziti i u odabranom skupu slika. Za ovu primjenu vrlo je bitno što točnije odrediti buduće lokacije objekata, primjerice pješaka, jer pri velikim brzinama često neće biti dovoljno reagirati u trenu kada se pojavi opasnost. Pravovremenim predviđanjem opasnosti moguće je reagirati i prije njene pojave. U literaturi postoji nekoliko distinktnih pristupa prognoziranju koje razlikujemo po vrsti ulazne odnosno izlazne reprezentacije [2]:

- iz slike u sliku (eng. *image to image*; I2I)
- iz semantičkih predikcija u semantičke predikcije (eng. *semantics to semantics*; S2S)
- iz optičkog toka u optički tok (ili iz pomaka u pomak) (eng. *motion to motion*; M2M)
- iz značajki u značajke (eng. *feature to feature*; F2F)

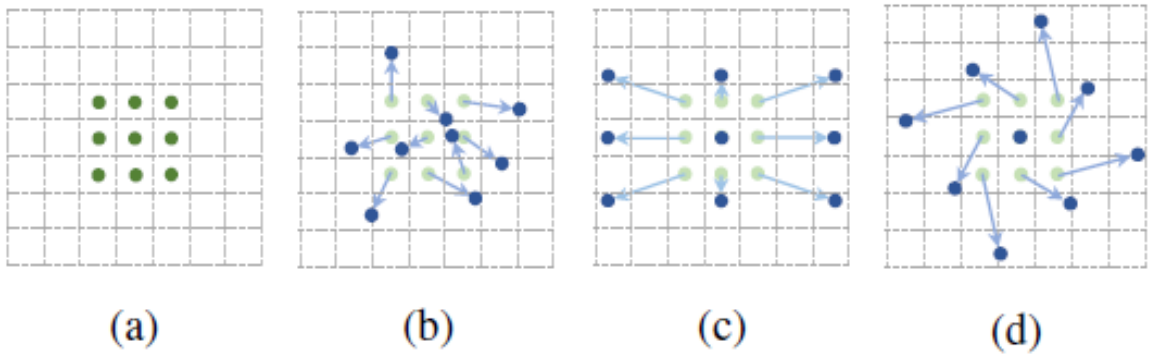
Naš pristup semantičkom prognoziranju oslanja se na značajke prognozirajući ih direktno (F2F) te na prognoziranje pomaka iz značajki (F2M) koji kombinira elemente F2F i M2M prognoziranja.

2.1. Deformabilne konvolucije

Konvolucijske arhitekture posljednje su desetljeće vrlo popularne u području računalnog vida. Međutim, predodređena pravilna struktura konvolucijske jezgre onemogućuje modeliranje geometrijskih transformacija (skaliranje, rotacija, deformacija, itd.) objekata koje nisu već viđene [10]. Ovaj problem obično pokušavamo umanjiti učenjem modela uz augmentaciju podataka koja uključuje zadane geometrijske transformacije. Ipak, ovim zaobilaznim putem ne možemo anticipirati sve moguće transformacije u kojima bi se mogao pojaviti objekt te uvodimo dodatni zahtjev za kapacitetom modela kako bismo mogli naučiti sve viđene transformacije. U radu [10] predloženo rješenje ovog problema je zamjena jezgre pravilnog te predodređenog oblika deformabilnom jezgrom, a takav sloj naziva se

deformabilna konvolucija. Deformabilna jezgra prilagođava se sadržaju ulaznog tenzora [2] što olakšava modeliranje transformiranih objekata. Lokacija uzorkovanja svakog prostornog elementa jezgre pomaknuta je od svojeg podrazumijevanog mjesta na diskretnoj rešetci za predviđeni nediskretni pomak [2].

Slika 2.1 ilustrira neke od mogućih izgleda deformabilne jezgre. Pod (a) je prikazana standardna konvolucijska jezgra na diskretnoj rešetci, dok se u slučajevima (b), (c) i (d) radi o deformabilnoj jezgri. Zeleni kružići predstavljaju podrazumijevane lokacije uzorkovanja pravilne jezgre, plave strelice su pomaci u odnosu na podrazumijevane lokacije koje dobivamo na temelju ulaza, a plavi kružići su stvarne nediskretne lokacije uzorkovanja deformabilne jezgre. Slučajevi (c) i (d) prikazuju kako ovakvom jezgrom možemo modelirati i pravilne geometrijske transformacije gdje jezgra (c) predstavlja deformaciju specijaliziranu za horizontalno skaliranje, a (d) za rotaciju (i skaliranje) objekta.

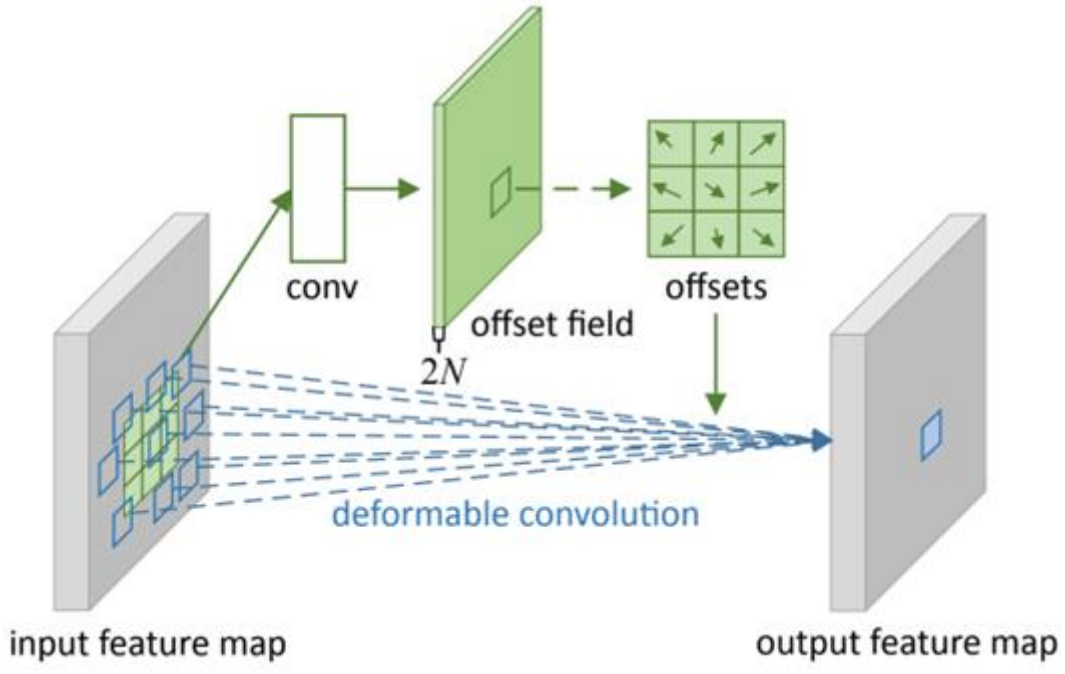


Slika 2.1: Mogući oblici deformabilne jezgre. Zeleni kružići predstavljaju podrazumijevane lokacije uzorkovanja pravilne jezgre, plave strelice su pomaci u donosu na podrazumijevane lokacije koje dobivamo na temelju ulaza, a plavi kružići su stvarne nediskretne lokacije uzorkovanja deformabilne jezgre. Pod (a) prikazana je pravilna konvolucijska jezgra veličine 3×3, dok su u slučajevima (b), (c) i (d) ilustrirane različite deformirane jezgre. Slika preuzeta iz [10].

Deformabilne konvolucije imaju potencijal riješiti još jedan problem konvolucijskih mreža. Receptivno polje svih jezgara istog sloja podjednako je i unaprijed određeno u klasičnim konvolucijskim arhitekturama. To je osobito nepoželjno u krajnjim slojevima kojima želimo modelirati semantiku [10] te otežava modeliranje objekata varijabilnih veličina (ili udaljenosti na slici). Deformabilne konvolucije uvode određen stupanj fleksibilnosti pri

čemu svaka jezgra istog sloja može imati različito receptivno polje te se prilagoditi kako velikim, tako i malim objektima.

Prostorni pomaci lokacija uzorkovanja jezgre regresiramo na temelju prethodnih reprezentacija modela te oni ovise o trenutnom položaju konvolucijske jezgre, odnosno o sadržaju reprezentacija s tog položaja [2]. Pomake lokacija jezgre računamo dodatnim konvolucijskim slojem kao što je prikazano slikom 2.2. Konvolucija zadužena za pomake (eng. *offset convolution*) na ulazu prima reprezentacije, a na izlazu predaje $2 \cdot k^2$ pomaka za svaku poziciju gdje k predstavlja veličinu deformabilne jezgre. Broj od $2 \cdot k^2$ pomaka proizlazi iz činjenice da su pomaci dvodimenzionalni pa za svaku lokaciju u jezgri, kojih ima k^2, trebamo 2 pomaka. Konačno, ovom konvolucijom dobivamo polje pomaka (eng. offset field) dimenzija $C \times H \times W \times 2k^2$, prikazano zelenom bojom na slici 2.2. Polje pomaka u sljedećem koraku predajemo deformabilnoj konvoluciji. Budući da pomaci nisu diskretni, reprezentacije koje uzorkuje deformabilna jezgra dobivamo bilinearnom interpolacijom [10]. Cijeli postupak može se učiti s kraja na kraja propagacijom pogreške unatrag (eng. *backpropagation*).



Slika 2.2: Deformabilna kovolucija. Pomaci se dobivaju za svaki položaj u reprezentacijama po kojemu se pomiče jezgra. Polje pomaka izračunava zaseban konvolucijski sloj. Slika preuzeta iz [10].

2.2. Korelacijski sloj

Korelacijski postupci već se dulje vrijeme koriste za izračun optičkog toka. U posljednje vrijeme najprecizniji modeli za procjenu optičkog toka zasnovani su na dubokom učenju [2]. Arhitekture poput FlowNeta [11] koriste konvolucijske slojeve za ekstrakciju reprezentacija nad kojima se računaju korelacije. Preostali konvolucijski slojevi dubokog modela služe za pronalazak korespondencija odnosno procjenu optičkog toka. Korelacijski sloj sastavni je dio prognostičkog modela F2MF [1][2] čiji se modul F2M oslanja na korelacijske značajke za predviđanje gustog polja vektora pomaka.

Naša osnovna implementacija modela F2MF koristi javno dostupan repozitorij za izračun korelacija namijenjen za korištenje uz biblioteku PyTorch. Sloj podržava propagaciju pogreške unatrag te efikasno izvršavanje na grafičkim karticama implementacijom u programskom jeziku CUDA. Kao i u [1][2], naš korelacijski sloj sadrži dodatan konvolucijski sloj koji služi za prilagođavanje reprezentacija za izračun korelacija. Ta konvolucija primjenjuje se zasebno na reprezentacije iz svakog vremenskog trenutka te sadrži dijeljene težine. Posljednji korak prije izračuna korelacija je normaliziranje reprezentacija. Korelacije računamo usporedbom isječaka veličine 1×1 radi smanjenja računske složenosti. Globalna usporedba isječaka (svaki sa svakim) je računalno skupa pa pretragu ograničavamo na lokalno susjedstvo oko trenutne lokacije [2]. U osnovnoj implementaciji lokalno susjedstvo je u obliku kvadrata veličine $k = 9$. Budući da naš model koristi reprezentacije iz četiri vremenska trenutka, primjenom korelacijskog sloja dobivamo tri para prostorno-vremenskih korelacijskih koeficijenata iz lokalnog susjedstva veličine $k \times k$ [2]. Dobivene korelacije predajemo sljedećim slojevima kao jedan tenzor dimenzija $B \times (T-1) \cdot k^2 \times H \times W$, gdje T označava broj vremenskih trenutaka.

2.3. Model F2F

Arhitektura F2F (ili DeformF2F) iz rada [9] osmišljena je za predviđanje semantičkih značajki, prvenstveno za primjenu u autonomnim vozilima. Glavna značajka arhitekture je da koristi samo slojeve deformabilnih konvolucija. Autori ističu kako je prognoziranje značajki iz značajki zadatak geometrijskog transformiranja reprezentacija, a ne semantičkog raspoznavanja, budući da su i ulazne i izlazne reprezentacije na istoj semantičkoj razini. Nadalje, obične konvolucije nemaju mogućnost naučiti geometrijske transformacije zbog predodređenog oblika rešetke za uzorkovanje [9]. Autori smatraju kako pomaci lokacija

uzorkovanja koji se mogu naučiti dobro odgovaraju zadatku koji zahtjeva modeliranje dinamike objekata u promatranim trenutcima (slikama).

Prva deformabilna konvolucija arhitekture F2F ima najviše ulaznih mapa značajki jer stapa reprezentacije iz svih ulaznih slika. Radi računske složenosti njena je jezgra veličine 1×1, dok svi ostali slojevi koriste 128 mapa značajki i jezgre veličine 3×3 [9]. U radu [9] ispitani su modeli s pet i osam slojeva koje ćemo dalje označavati s F2F-5 i F2F-8. Rad [2] ističe kako je velika prednost F2F modela što ima sposobnost „zamišljanja" značajki, odnosno što model regresira značajke i u novootkrivenim dijelovima scene. Zamislimo, na primjer, situaciju u kojoj se ispred kamere na sve četiri ulazne slike nalazi kamion s prikolicom koji zakriva veliki dio scene. Ako se kamion kreće možemo predvidjeti kako će se u prognoziranom trenutku pojaviti novi dio scene koji je prije bio zakriven. F2F model će u tom slučaju prognozirati značajke u novootkrivenom dijelu scene iako nema prijašnju informaciju o tome što bi se u stvarnosti tamo moglo nalaziti. U sljedećem ćemo poglavlju razmatrati F2F kao dio kompleksnijeg F2MF modela.

2.4. Modul F2M

Modul F2M sastavni je dio arhitekture F2MF predstavljene radom [1]. Pretpostavka modula F2M je da se budućnost u potpunosti može objasniti geometrijskom transformacijom iz prošlosti [2]. Kao i model F2F, F2M se također sastoji od deformabilnih slojeva. Posljednji deformabilni sloj predviđa gusto polje vektora pomaka za reprezentacije iz svakog vremenskog trenutka, kojih u našem slučaju ima četiri. Tako na izlazu modula dobivamo tenzor dimenzija $B \times T \cdot 2 \times H \times W$, gdje T označava broj vremenskih trenutaka (kod nas je $T = 4$). Broj kanala jednak je $T \cdot 2$ jer svaki pomak sadrži horizontalnu i vertikalnu komponentu. Buduće reprezentacije dobivamo deformacijom reprezentacija iz odgovarajućeg trenutka s regresiranim pomacima iz tog trenutka. Tako, zapravo, isti trenutak iz budućnosti pokušavamo objasniti deformacijom značajki iz T različitih trenutaka u prošlosti [2]. Dobivene procjene u posljednjem se koraku miješaju u skladu s predviđenom težinskom sumom, koju ćemo objasniti u sljedećem poglavlju.

Rad [1] razmatra korištenje unaprijedne i unatražnje deformacije prošlih reprezentacija. Niti jedan od ova dva pristupa nije bez mana, no kod unaprijedne transformacije pojavljuje se problem nastajanja „rupa" u prognoziranim reprezentacijama gdje nema predviđenih pomaka. Iz tog razloga češće se koristi unatražnja transformacija.

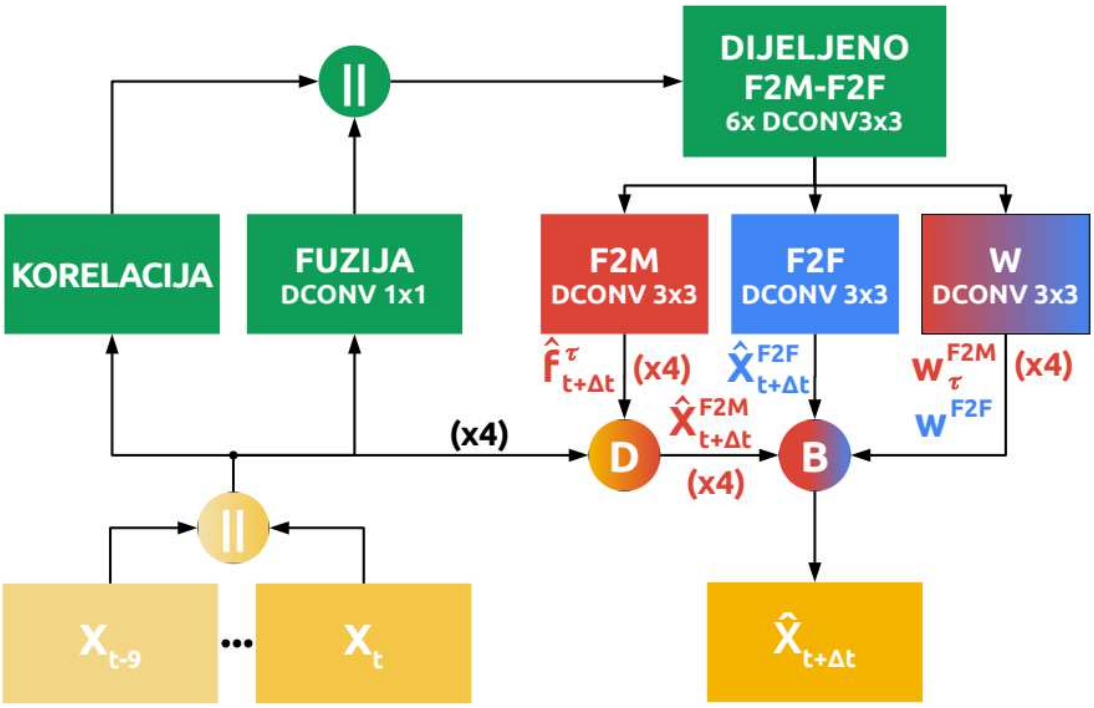
2.5. Model F2MF

Arhitektura F2MF [1][2] objedinjuje izravno prognoziranje semantičkih značajki (F2F) i transformaciju značajki uz regresiju pomaka (F2M). Motivacija iza ovog pristupa bila je kombinirati prednosti F2F i F2M modula te iskoristiti njihove sličnosti za efikasno dijeljenje reprezentacija.

Na ulaz F2MF predajemo konkatenirani (po kanalima) tenzor reprezentacija iz T prošlih vremenskih trenutaka. Nad takvim se tenzorom računaju korelacijski koeficijenti, za koje je zaslužan korelacijski modul, te kombinirana reprezentacija iz prošlih trenutaka. Kombiniranu reprezentaciju izračunava jedan deformabilni sloj koji zbog efikasnosti ima jezgru veličine 1×1. Mi sloju za kombiniranje reprezentacija (također ga zovemo i fuzija) postavljamo broj izlaznih kanala na 128. Korelacijski modul i fuzijski sloj označeni su zelenim pravokutnicima na lijevom dijelu slike 2.3. Korelacijske koeficijente i kombinirane reprezentacije naknadno konkateniramo po kanalima te predajemo dijeljenom modulu. Dijeljeni modul sastoji se od šest deformabilnih slojeva s jezgrom veličine 3×3. Prvi deformabilni sloj smanjuje broj mapi značajki na predodređenu vrijednost. Ta vrijednost odgovara ulaznom i izlaznom broju kanala svih ostalih slojeva dijeljenog modula, a mi ju postavljamo na 256. Na izlazu dobivamo dijeljenu reprezentaciju za prognoziranje modulima F2F i F2M.

Moduli F2F i F2M su „plitki“ jer sadrže samo jedan deformabilni sloj zadužen za specijalizaciju značajki iz dijeljene reprezentacije za jednu od te dvije primjene. Analogno s multimodalnim arhitekturama, dijeljene deformabilne slojeve možemo smatrati „vratom“ F2MF arhitekture, dok su F2F (plavi pravokutnik na slici 2.3) i F2M (crveni pravokutnik na slici 2.3) „glave“ modela. Deformabilni sloj F2F modula pretvara broj kanala na početni broj mapa značajki iz okosnice, dok modul F2M regresira polje pomaka s $T \cdot 2$ kanala. Polje pomaka primjenjujemo na semantičke značajke iz ulaznih trenutaka, čime dobivamo T deformiranih ulaznih reprezentacija. Ostaje pitanje kako iz T prognoziranih reprezentacija dobivenih iz F2M modula i jednih reprezentacija regresiranih iz F2F modula dobiti konačne prognozirane značajke iz budućnosti. U tu svrhu uvedena je još jedna „težinska glava“ označena plavo-crvenim pravokutnikom i slovom W na slici 2.3. Ona se također sastoji od jednog deformabilnog sloja čiji je zadatak predvidjeti težine kojima će se miješati prognozirane značajke [2]. U našem će slučaju težinski sloj dati pet mapa težina na razini piksela. Prognozirane se značajke miješaju sukladno težinama aktiviranim funkcijom

softmax na razini piksela. Težinski sloj omogućuje naučenu specijalizaciju modula F2F i F2M za određene dijelove ulazne scene. Rad [2] tako, na primjer, sugerira specijalizaciju F2F modula za novootkrivene dijelove scene.



Slika 2.3: Detaljan prikaz F2MF modela. Konkatenirani korelacijski koeficijenti i kombinirane semantičke značajke predaju se dijeljenim slojevima. Dijeljene reprezentacije koriste se za prognoziranje 2D pomaka u modulu F2M, regresiranje budućih značajki u modulu F2F te predviđanje težina. Deformiranjem semantičkih značajki iz prošlih trenutaka dobivamo četiri para predviđenih značajki. Ukupno pet parova značajki miješa se sukladno težinama kako bismo dobili konačne prognozirane reprezentacije iz budućnosti. Slika preuzeta iz [2].

Prilikom učenja F2MF modela, naša funkcija gubitka ima tri komponente sukladno radu [1]. Gubitak se računa kao srednja kvadratna pogreška između reprezentacija dobivenih primjenom okosnice na stvarnu sliku iz budućnosti i prognoziranih značajki iz različitih stadija. Tri komponente gubitka računaju se temeljem konačnih prognoziranih reprezentacija sveukupnog F2MF modela te prognoza F2M i F2F modula. Važna napomena je da, za potrebe izračuna gubitka, računamo kombiniranu reprezentaciju iz četiri para značajki dobivenih F2M modulom temeljem četiri od pet prognoziranih mapa težina. Sve tri komponente doprinose podjednako ukupnom gubitku.

3. Generativno modeliranje RGB slika

Popularnost generativnih modela posljednjih je godina znatno narasla, posebice zahvaljujući napretku arhitektura za uvjetno generiranje slika i teksta. U ovom ćemo radu koristiti generativni model za drugačiji zadatak, a to je rekonstrukcija slika. Rekonstrukcija uključuje transformaciju ulaznih podataka u latentnu reprezentaciju od koje je nazad, drugom transformacijom, potrebno dobiti izlaz što vjerniji ulazu. Jedna zanimljiva primjena rekonstrukcije je kompresija slika, kojom sažete latentne reprezentacije često mogu vjerno obuhvatiti bitne informacije sa slike.

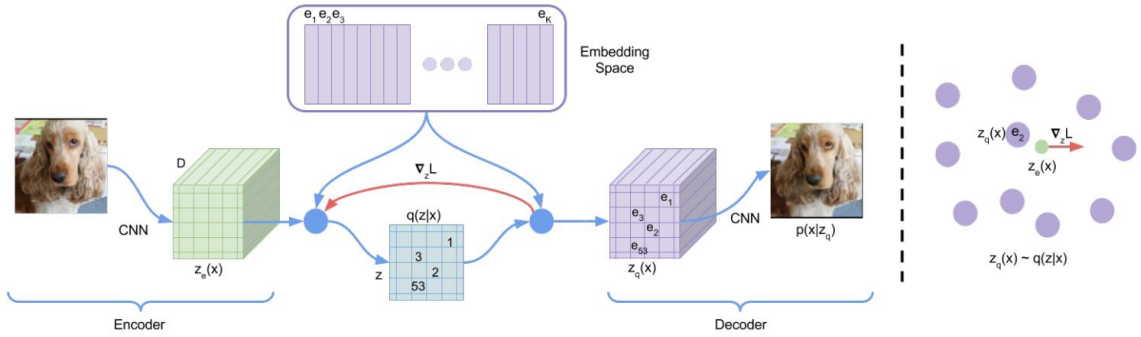
U ovom radu razmotrit ćemo pristup prognoziranja RGB slika u videu prognoziranjem latentnih reprezentacija iz generativnog modela. Odlučili smo koristiti predtrenirani model čije su latentne reprezentacije slične rezolucije kao i prognozirane segmentacijske značajke iz prethodnog poglavlja. To je suzilo izbor na arhitekturu VQGAN [12] (eng. *vector quantised generative adversarial network*) čije su reprezentacije (u određenim konfiguracijama) rezolucije /16. Arhitektura VQGAN-a zapravo je identična arhitekturi VQ-VAE [13] (eng. *vector quantised-variational autoencoder*) uz izmijenjen pristup treniranju modela.

Varijacijski autoenkoder sastoji se od tri [13] dijela: (i) enkodera koji parametrizira aposteriornu distribuciju $q(z|x)$ latentne varijable z u ovisnosti o ulaznim podacima x, (ii) apriorne distribucije $p(z)$ te (iii) dekodera koji modelira distribuciju $p(x|z)$. Najčešće se za prior i posterior uzima Gaussova (normalna) distribucija, dok su kod VQ-VAE ove distribucije kategoričke. Reprezentacije ovih kategorija su vektori u latentnom prostoru koji su indeksirani u tablici (rječniku) ugrađivanja (eng. *embeddings*). Latentni prostor ugrađivanja definiran je s $\mathrm{e} \in R^{\mathrm{K}\times\mathrm{D}}$, gdje je K veličina rječnika ugrađivanja, a D dimenzionalnost svakog latentnog vektora ugrađivanja e_i [13]. Nadalje, rječnik se sastoji od ukupno K vektora ugrađivanja $e_i \in R^{\mathrm{D}},\ i \in 1,2,\dots,\mathrm{K}$.

Zaključivanje VQ-VAE-a započinje prolaskom ulaza x kroz enkoder, čime dobivamo izračunate reprezentacije $z_e(x)$. Kvantizacija se provodi pronalaskom najbližeg susjeda $e_j \in R^{\mathrm{D}}$ (iz dijeljenog rječnika ugrađivanja) za svaku dobivenu reprezentaciju $z_e(x)$ kao u izrazu 3. Pri tome se uzima L_2 udaljenost kao kriterij pretrage najbližeg susjeda.

$$z_q(x) = e_k, \qquad k = arg\ min_j \left\| z_e(x) - e_j \right\|_2 \tag{3}$$

Sada je jasno kako latentne reprezentacije slike, nakon kvantizacije, možemo vrlo sažeto zapisati kao matricu indeksa (iz rječnika ugrađivanja) dimenzija $H/16 \times W/16$, gdje H i W označavaju dimenzije ulazne slike. Konačno, dekoder rekonstruira sliku iz kvantiziranih vektorskih reprezentacija e_k. Arhitektura i proces zaključivanja VQ-VAE modela također su prikazani i slikom 3.1. Lijevi dio prikaza predstavlja rekonstrukciju slike, koja se sastoji od tri stadija: prolaska kroz enkoder čime se dobiju reprezentacije, kvantizacije latentnih reprezentacija uz rječnik ugrađivanja te dekodiranje dobivenih reprezentacija. Desni dio prikaza odnosi se na pronalazak najbližeg susjeda $z_q(x)$ za ulazni vektor $z_e(x)$. Crvena strelica na lijevom dijelu prikaza označava kopiranje gradijenata od ulaza dekodera do izlaza enkodera. Ovaj pristup omogućava jednostavno učenje enkodera te zaobilazi kompleksniju kvantizacijsku operaciju.



Slika 3.1: Lijevi dio prikaza ilustrira arhitekturu VQ-VAE modela te prikazuje proces rekonstrukcije slike u tri odvojena stadija. Desni dio prikaza odnosi se na drugi stadij (kvantizacija), gdje se ulazni latentni vektori zamjenjuju najbližim susjedima iz rječnika ugrađivanja. Slika preuzeta iz [13].

Gubitak VQ-VAE modela sastoji se od tri komponente. Prva komponenta je rekonstrukcijski gubitak, a VQ-VAE koristi L_2 udaljenost kao mjeru pogreške rekonstrukcije u odnosu na ulaznu sliku. Druge dvije komponente gubitka imaju zadatak pritegnuti vektore ugrađivanja e_i što bliže izlazima dekodera $z_e(x)$ te spriječiti udaljavanje izlaza dekodera od ugrađivanja iz rječnika [13]. VQGAN uvodi suparničko treniranje s diskriminatorom koji se primjenjuje zasebno na dijelove (prozore) slike. Kao i u ostalim modelima zasnovanima na suparničkom gubitku (eng. *generative adversarial network*), zadaća ovakvog diskriminatora je prepoznavanje stvarnih od generiranih slika. VQGAN također zamjenjuje rekonstrukcijski L_2 gubitak perceptualnim gubitkom (eng. *perceptual loss*). Obje promjene uvedene su s ciljem generiranja vizualno ugodnijih slika.

4. Detalji izvedbe

4.1. Podatkovni skup Cityscapes

Cityscapes je skup slika namijenjen prvenstveno za semantičku segmentaciju urbanih uličnih scena. Skup sadrži ukupno 30 semantičkih klasa, a mi koristimo podskup od 19 klasa. Cityscapes sastoji se od 5000 sekvenci snimljenih u 50 gradova u dobrim vremenskim uvjetima. Podatkovni skup sadrži 2975 sekvenci za treniranje, 500 za validaciju i 1525 za testiranje, s time da oznake skupa za testiranje nisu javno dostupne. Slike su veličine 1024×2048 piksela. Svaka se sekvenca sastoji od ukupno 30 sličica, a od njih samo 20. slika dolazi u paru s detaljnim semantičkim oznakama. Točnije, 19 slika prije i 10 slika nakon ciljne slike su neoznačene. Cityscapes sadrži i grube oznake na većem broju slika te panoptičke oznake, oznake instanci, podatke o dubini i mnoge druge metapodatke koje mi u ovom radu nećemo koristiti.

Slika 4.1: Primjer slike i oznaka iz podatkovnog skupa Cityscapes. Originalna RGB slika preklopljena je s obojenim semantičkim oznakama.

Za osnovno treniranje prognostičkih modela koristit ćemo slike iz četiri vremenska trenutka koje su snimljene prije ciljne slike. Ciljna slika bit će 20. sličica iz sekvence koja sadrži fine semantičke oznake. Ulazne slike snimljene su u razmaku od tri sličice. Razlikujemo kratkoročno (tri sličice unaprijed, odnosno 0.18 sekundi) i srednjeročno prognoziranje (devet

sličica unaprijed, odnosno 0.54 sekundi) [2]. Također ćemo koristiti i prošireni skup podataka za treniranje gdje ćemo, umjesto jednog, uzeti četiri uzorka iz svake sekvence. To je moguće jer naše prognostičke modele učimo nenadzirano i potpuno neovisno o semantičkom modelu pa nam semantičke oznake u ovom koraku nisu potrebne. Tako će ciljne slike iz jedne sekvence biti 20., 23., 26. i 29. sličica. Kako bismo ubrzali treniranje prognostičkih modela spremili smo značajke dobivene okosnicom iz svih slika potrebnih za treniranje i evaluaciju modela. Značajke smo spremili na SSD disk što potpuno uklanja potrebu za korištenje okosnice semantičkog modela, dok se modul za naduzorkovanje koristi prilikom evaluacije.

4.2. Korištene metrike

Za evaluaciju semantičke segmentacije koristimo mjeru omjer presjeka i unije, odnosno usrednjeni omjer presjeka i unije (eng. *mean intersection over union*, mIoU) budući da želimo dobiti jedinstvenu mjeru neovisno o klasama objekata. Istu mjeru koristimo i pri semantičkom prognoziranju s time da dodajemo i podskup mjere mIoU koja se računa samo nad klasama pokretnih objekata. U slučaju skupa Cityscapes to su: osoba, vozač, automobil, kamion, autobus, vlak, motocikl i bicikl. Ta mjera naziva se mIoU-MO (eng. mIoU – *moving objects*) te je iznimno bitna za praćenje sposobnosti modeliranja kretanja.

Za prognoziranje RGB sličica u videu koristimo mjeru srednje kvadratne pogreške (eng. *mean square error*, MSE) te se oslanjamo na vizualnu procjenu kvalitete prognoziranih rekonstrukcija.

4.3. Programska izvedba

Cjelokupni kod u sklopu ovog rada napisan je u programskom jeziku Python. Postoji mnogo biblioteka napisanih za Python koje sadrže alate za gotovo sve grane računarstva. Posebno su zastupljene biblioteke za strojno učenje, što je i učinilo Python najzastupljenijim jezikom u ovom području istraživanja. Kod za ovaj rad napisali smo kao proširenje repozitorija swiftnet, a koristili smo još i sljedeće biblioteke i repozitorije:

- PyTorch – strojno i duboko učenje uz potporu za automatsku diferencijaciju s podrškom za grafičke kartice
- Torchvision – proširenje PyTorcha za računalni vid
- NumPy – efikasno računanje s multidimenzionalnim podacima (tenzorima)
- Pillow – manipuliranje slika
- OpenCV – mnoge primjene u računalnom vidu, prikazivanje slika
- Matplotlib – crtanje te prikazivanje slika i grafova
- tqdm – prikaz progresne trake
- swiftnet – repozitorij s implementacijom segementacijskog modela SwiftNet
- PyTorch Correlation module – repozitorij za računanje korelacija s efikasnom implementacijom za izvođenje na grafičkim karticama
- Taming transformers – repozitorij s implementacijom mnogih generativnih modela u PyTorchu iz članka [12].

5. Eksperimenti

U ovom poglavlju opisat ćemo provedene eksperimente i komentirati dobivene rezultate. Prije eksperimentiranja s prognoziranjem bilo je potrebno naučiti segmentacijski model čije će se značajke prognozirati. Treniranje modela za semantičku segmentaciju opisano je u potpoglavlju 5.1. Sljedeće potpoglavlje postavlja osnovicu usporedbe uvodeći eksperimente s našom implementacijom prognostičkih modela iz literature. Potpoglavlja 5.3 i 5.4 proučavaju razne modifikacije ovih arhitektura s ciljem poboljšanja točnosti prognoza. Potpoglavlje 5.5 proučava utjecaj modifikacija na brzinu zaključivanja, odnosno uspoređuje računalnu složenost naših modela. Konačno, eksperimenti iz posljednjeg potpoglavlja ispituju mogu li se predstavljeni modeli iskoristiti za prognoziranje RGB slika u videu.

5.1. Semantička segmentacija

Broj zadržanih preskočnih veza u piramidalnom SwiftNetu direktno utječe na dimenzionalnost prognoziranih značajki jer se prognoziranje odnosi na posljednji blok naduzorkovanja koji ima preskočne veze. Pritom se svakom dodatnom preskočnom vezom dimenzije prognoziranih značajki uvećavaju za faktor 2×. Ispitali smo više konfiguracija arhitekture SwiftNet mijenjajući broj piramidalnih slojeva te zadržavajući N početnih preskočnih veza.

Segmentacijski model treniran je kao i u doktorskoj disertaciji dr.sc. Josipa Šarića [2]. Kralježnica (eng. *backbone*) SwiftNeta je ResNet-18 predtreniran na skupu ImageNet. Model smo trenirali 250 epoha optimizacijskim postupkom ADAM uz varijabilnu stopu učenja između $4 \cdot 10^{-4}$ i $1 \cdot 10^{-6}$ te veličinu grupe 16. Skup podataka umjetno smo povećali korištenjem standardnih augmentacijskih tehnik a. Konačni uzorci iz augmentiranih slika bili su veličine 768×768 piksela.

Iz tablica 5.1 i 5.2 vidljivo je da micanje preskočnih veza negativno utječe na segmentacijske performanse modela. Tablica 5.1 odnosi se na piramidalni SwiftNet s dvije razine prognoziranja te prikazuje veću razliku u performansama modela s jednom i dvije preskočne veze u odnosu na razliku kod modela s dvije i tri veze. Prve preskočne veze odnose se na veze s modulima za naduzorkovanje u kojima se kombiniraju značajke najnižih rezolucija. Iz ovih se rezultata može zaključiti kako upravo prve preskočne veze najviše utječu na kvalitetu segmentacijskog izlaza modela.

Tablica 5.1: Piramidalni SwiftNet s okosnicom ResNet-18 predtreniranom na ImageNetu. Konfiguracije arhitekture sadrži dvije razine piramide, a broj zadržanih početnih veza varira u tablici. Eksperimenti su provedeni na validacijskom skupu Cityscapes uz originalnu slikovnu rezoluciju.

Model	mIoU (%)
SwiftNet, 1 preskočna veza	72.16
SwiftNet, 2 preskočne veze	73.66
SwiftNet, 3 preskočne veze	73.94

Segmentacijske predikcije trorazinskog piramidalnog SwiftNeta prikazane su u tablici 5.2, gdje se važnost prvih preskočnih veza još više ističe. Dodavanjem treće preskočne veze performanse modela poboljšavaju se za gotovo dva postotna boda mjere mIoU. Nasuprot tome, dodavanje još i četvrte i pete preskočne veze ima znatno slabiji utjecaj na preciznost modela budući da je dobiveno poboljšanje manje od jednog postotnog boda mIoU. Ovdje je bitno napomenuti kako puni trorazinski SwiftNet ima ukupno pet preskočnih veza, dok puni dvorazinski model ima četiri veze. Razlika u ukupnom broju preskočnih veza kod ove dvije konfiguracije SwiftNeta rezultira odudaranjem dimenzija značajki nakon što oba modela odrade isti broj koraka naduzorkovanja. Značajke dvorazinskog modela uvijek će biti $2\times$ veće u odnosu na značajke iz trorazinskog modela koje su prošle isti broj koraka naduzorkovanja. Zato su značajke trorazinskog modela s tri preskočne veze ekvivalentne značajkama s dvije preskočne veze kod dvorazinskog modela, a njihove dimenzije iznose $\mathrm{B} \times \mathrm{C} \times \mathrm{H}/16 \times \mathrm{W}/16$ ($\mathrm{B} \times \mathrm{C} \times 64 \times 128$ za slike iz skupa Cityscapes koje su rezolucije 1024×2048 piksela) gdje B označava veličinu grupe, C broj kanala, a H i W visinu i širinu. Ako isto pravilo primijenimo na modele s jednom preskočnom vezom manje, nakon naduzorkovanja s preskočnim vezama dimenzije značajki iznosit će $\mathrm{B} \times \mathrm{C} \times \mathrm{H}/32 \times \mathrm{W}/32$.

Tablica 5.2 Piramidalni SwiftNet (okosnica ResNet-18) s tri razine, broj početnih preskočnih veza varira u tablici. Eksperimenti su provedeni na skupu Cityscapes.

Model	mIoU (%)
SwiftNet, 2 preskočne veze	74.06
SwiftNet, 3 preskočne veze	75.95
SwiftNet, 5 preskočnih veza (bez modifikacija)	76.80

Eksperimentiranje s konfiguracijama SwiftNeta pokazalo je kako model s tri piramidalne razine postiže znatno bolje rezultate od modela s dvije razine te kako micanje preskočnih veza može imati negativan utjecaj na performanse. Budući da za semantičko prognoziranje koristimo značajke iz zadnjeg koraka naduzorkovanja s preskočnim vezama, potrebno je pažljivo odabrati koliko ćemo preskočnih veza koristiti te pritom uspostaviti balans između računalne složenosti i točnosti. Kako veličina tenzora značajki znatno utječe na efikasnost prognoziranja, u fokusu će nam biti značajke dimenzija 32×64 te 64×128.

5.2. Prognoziranje semantičkih značajki

Za osnovicu usporedbe (engl. *baseline*) prognozirali smo značajke trorazinskog SwiftNeta s dvije preskočne veze, kao što je opisano u poglavlju 5.1. Radi usporedbe, modele smo trenirali sukladno parametrima iz radova [1][2]. Glavna razlika je što se u navedenim radovima razmatra prognoziranje značajki jednorazinskog SwiftNeta bez preskočnih veza te se osim okosnice ResNet-18 koristi i mnogo veći model – DenseNet-121. Osim toga, jednorazinski SwiftNet također koristi i SPP modul (eng. *spatial pyramid pooling*). SPP usrednjuje značajke u rešetkama grublje rezolucije u odnosu na početne reprezentacije te time efektivno proširuje receptivno polje i uvodi globalni kontekst.

U ovom poglavlju, osim srednjeročnog (devet sličica unaprijed), prikazat ćemo i kratkoročno prognoziranja (tri sličice unaprijed). U sljedećim poglavljima fokus će biti na srednjeročnim rezultatima kako bi se smanjio broj eksperimenata.

F2F modeli iz tablice 5.3 trenirani su 160 epoha optimizatorom ADAM uz stopu učenja $5 \cdot 10^{-4}$ te veličinu grupe 12. Model F2MF treniran je istim parametrima, osim što smo koristili i kosinusno kaljenje za smanjivanje stope učenja do $1 \cdot 10^{-7}$. U oba slučaja modele smo učili bez augmentacije podataka.

Tablica 5.3: Rezultati kratkoročnog i srednjeročnog prognoziranja značajki trorazinskog SwiftNeta s prognostičkim modelima preuzetim iz radova [1][2] na validacijskom skupu Cityscapes. Oracle predstavlja točnost segmentacijskog modela uz poznatu sliku iz budućeg trenutka. Uz naše modele prikazana su i dva eksperimenta prognoziranja značajki jednorazinskog SwiftNeta iz literature, segmentacijska okosnica u prvom pristupu je ResNet-18 (kao i kod nas), dok drugi pristup koristi DenseNet-121 za ekstrakciju značajki.

	Kratkoročno (Δt=3)		Srednjeročno (Δt=9)	
	mIoU (%)	mIoU-MO (%)	mIoU (%)	mIoU-MO (%)
Oracle	74.06	73.39	74.06	73.39
F2F-5	64.54	62.72	51.13	46.60
F2F-8	65.36	63.64	51.60	47.02
F2MF	67.05	65.62	55.03	50.97
F2MF [1][2]	66.90	65.60	55.90	52.40
F2MF (DenseNet-121)[1][2]	68.70	66.80	56.80	53.10

Prvi redak tablice 5.3 predstavlja performanse segmentacijskog modela kada bi nam bila poznata RGB slika iz budućeg trenutka u kojem želimo prognozirati oznake. Takav model nazvat ćemo prorokom (eng. *oracle*). Ovo je nerealan scenarij i služi nam za uspostavljanje gornje granice performansi prognoziranja, odnosno očekujemo kako će rezultati prognoziranja uvijek biti lošiji od proroka.

Naš prorok postiže 74.06% mIoU te nadjačava jednorazinski SwiftNet iz rada [1] koji postiže 72.50% mIoU. Usprkos ovome, naši rezultati prognoziranja neočekivano su ipak nešto slabiji od onih iz rada [1], gdje točnost F2F-8 modela za srednjeročno prognoziranje iznosi 52.80% mIoU, a F2MF modela 55.90% mIoU. U našim eksperimentima F2F-8 i F2MF postižu 51.60% i 55.03% mIoU, a rezultati kratkoročnog prognoziranja prate ista opažanja. Smatramo kako je najvjerojatniji uzrok zašto prognostički modeli u našim eksperimentima postižu nešto slabiju točnost taj što korištena segmentacijska arhitektura nema SPP modul. SPP modul ugrađuje proširenu kontekstualnu informaciju u značajke što može biti važan prediktor za računanje pomaka deformabilnih slojeva. Jednorazinski SwiftNet s okosnicom DenseNet-121 iz rada [1] ima segmentacijsku točnost od 75.80% mIoU, a F2MF model koji prognozira ovakve značajke postiže najbolje srednjeročne prognostičke rezultate – 56.80% mIoU.

5.3. Modifikacije modela za prognoziranje

U ovom potpoglavlju razmotrit ćemo nekoliko modifikacija prognostičkih modela s ciljem poboljšanja točnosti. Zbog velikog broja eksperimenata i različitih konfiguracija prikazat ćemo samo rezultate srednjeročnog prognoziranja.

5.3.1. Odvojeni deformabilni sloj

Prva modifikacija namijenjena je prvom deformabilnom sloju prognostičkih modela s ciljem povećanja njegovog kapaciteta. Cilj ove modifikacije bio je omogućiti modelu da za svaki od vremenskih trenutaka s ulaza može odabrati različite lokacije s kojih će se kombinirati značajke pojedinog piksela. Ovo smo postigli razdvajanjem prvog deformabilnog sloja na četiri zasebne deformabilne konvolucije, gdje je svaka specijalizirana za jedan od četiri vremenska trenutka, te ćemo stoga ovakav sloj nazivati odvojeni deformabilni sloj.

Pomaci deformabilne jezgre računaju se na temelju značajki iz svih trenutaka, ali se konvolucija primjenjuje na značajke iz samo jednog vremenskog trenutka. Značajke iz svakog trenutka zadržavaju isti broj kanala prolaskom kroz odvojeni deformabilni sloj te ih je stoga bilo potrebno na neki način kombinirati kako bi se smanjio ukupni broj kanala. Ako značajke iz jednog trenutka, primjerice, imaju 128 kanala, onda će na ulaz deformabilnog sloja doći konkatenirani tenzor s 512 kanala, a na izlazu ćemo također dobiti tenzor s 512 kanala. U nastavku poglavlja razmotrit ćemo nekoliko pristupa kombiniranju konkateniranih značajki.

Najjednostavnije rješenje za kombiniranje značajki bilo je samo promijeniti sljedeći deformabilni sloj tako da prima 4× broj kanala (512 kod SwiftNeta), a na izlazu daje 1× kanala (128 kod SwiftNeta). Takav sloj možemo nazvati deformabilni sloj uskoga grla (eng. *bottleneck*), a on će biti identičan prvome sloju nemodificiranog F2F modela te će mu jezgra biti veličine 1×1. Slično tome, isprobano je i dodavanje uskog grla s običnom konvolucijom. Ideja iza korištenja obične konvolucije bila je da će, ako odvojeni deformabilni sloj izračuna korektne pomake jezgre, trebati izračunati značajke jednoga piksela na osnovi svih prijašnjih značajki samo tog piksela pri čemu bi nam mogla smetati nepravilna jezgra deformabilnih konvolucija. U praksi se, pak, pokazalo kako oba pristupa daju nešto lošije rezultate u odnosu na osnovicu usporedbe.

Sljedeći pristup bio je kombinirati značajke pomoću elementarnog zbrajanja značajki iz sva četiri trenutka. Ovaj pristup računski je puno efikasniji te izbacuje potrebu za učenjem sloja koji kombinira značajke. Umjesto zbrajanja, također bismo mogli uzeti i srednje vrijednosti značajki iz sva četiri trenutka pa smo sljedeće isprobali usrednjavanje reprezentacija. Konačno, razmotrili smo i težinsko miješanje značajki inspirirano F2MF modelom iz rada [1]. Težine se dobivaju na razini piksela dodatnim deformabilnim slojem s četiri izlazna kanala i aktivacijskom funkcijom softmax. Izlaz iz sloja je težinska suma u kojoj se svaka izlazna značajka dobiva zbrajanjem značajki iz prošlih trenutaka pomnoženih s dobivenim težinama za trenutni piksel. Ovime modelu dajemo slobodu odabira važnosti značajki iz svakog vremenskog trenutka na razini piksela.

Eksperimenti su pokazali kako najjednostavniji pristup, a to je zbrajanje značajki, daje dobre rezultate. Zbrajanjem značajki dobiveno je blago povećanje točnosti od 0.13% mIoU. Najveće poboljšanje točnosti dobili smo, pak, težinskim miješanjem te ono iznosi 0.31% mIoU. Razlika ovakvog težinskog miješanja i onog korištenog u F2MF modelu iz rada [1] je što F2MF ima priliku prilagoditi značajke nad kojima se izračunavaju težine, dok mi težine dobivamo direktno iz semantičkih značajki. Dodavanjem slojeva (kapaciteta) težinskom modulu mogli bismo dobiti specijalizirane značajke, ali pod cijenu dodatne računske složenosti, što se kosi s efikasnim pristupom dijeljenja značajki. Tablica 5.4 izdvaja dva rezultata koja donose povećanje točnosti zadebljanim znamenkama, dok su ostali pristupi kombiniranja značajki rezultirali blagom redukcijom točnosti.

Tablica 5.4: Metode kombiniranja značajki nakon odvojenog deformabilnog sloja. Kao osnovica usporedbe koristi se nemodificirani F2F-8. Kod ostalih modela samo je prvi deformabilni sloj odvojen, dok je preostalih sedam slojeva identično kao i u F2F-8 modelu. Eksperimenti prikazuju prognoziranje semantičkih značajki dobivenih iz skupa Cityscapes, pri rezoluciji 32×64 ($/32$).

	Srednjeročno	
	mIoU (%)	mIoU-MO (%)
Bez modifikacija	51.60	47.02
Deformabilna konvolucija (512 → 128 kanala)	51.16	46.59
Konvolucija (512 → 128 kanala)	50.91	46.38
Zbrajanje značajki	**51.73**	**47.41**
Usrednjavanje značajki	51.11	46.05
Težinsko miješanje	**51.91**	**47.56**

Dodatno smo eksperimentirali i s nekim drugim manjim varijacijama ovih pristupa koji nisu prikazani u tablici 5.4, poput mijenjanja veličine konvolucijskih jezgri ili kombiniranja zbrajanja i konvolucije, no njima nisu dobiveni zanimljivi rezultati. Treba naglasiti kako je varijanca u ovim eksperimentima poprilično velika te je često teško razlučiti donosi li određena modifikacija poboljšanje/pogoršanje performansi ili se samo radi o statističkoj pogrešci. Stoga, kako bismo smanjili posljedice ovoga problema, svi su eksperimenti pokrenuti nekoliko puta uzastopno.

5.3.2. Prognoziranje značajki veće rezolucije

Sljedeća modifikacija više se odnosi na značajke nego na sami model. Naime, dodavanjem preskočne veze semantičkom modelu kao što je opisano u poglavlju 5.1 efektivno povećavamo visinu i širinu tenzora značajki za faktor $2\times$. Tako ćemo, umjesto tenzora dimenzija $\mathrm{B} \times \mathrm{C} \times 32 \times 64$, prognozirati tenzor dimenzija $\mathrm{B} \times \mathrm{C} \times 64 \times 128$. Ovime će računska složenost pri zaključivanju porasti za faktor $4\times$ iako nismo mijenjali prognostički model. Značajke veće rezolucije mnogo su finije od značajki rezolucije 32×64 što će, kao što ćemo prikazati, povoljno utjecati na performanse modela.

U tablici 5.5 prikazano je prognoziranje F2F modelom, a značajke su dobivene trorazinskim SwiftNetom s dvije i tri preskočne veze. Ujedno, zbog veće računske složenosti, isproban je i manji F2F-5 model. Dobiveni rezultati pokazuju da točnost oba modela osjetno raste koristeći značajke veće rezolucije. Tako performanse F2F-5 modela rastu za 1.28 postotnih bodova mjere mIoU, dok je kod F2F-8 modela rast 1.80 postotnih bodova mIoU. Veći rast performansi kod dubljeg prognostičkog modela sugerira kako je veće apsolutne pomake objekata u pikselima lakše modelirati s više deformabilnih slojeva. Budući da su objekti dva puta veći, modelu je teže postići receptivno polje veliko poput onoga kada se koriste značajke niže rezolucije.

Tablica 5.5: Prognoziranje značajki dimenzija 32×64 i 64×128 dobivenih trorazinskim SwiftNetom s dvije i tri preskočne veze. Performanse modela F2F-5 i F2F-8 znatno rastu koristeći značajke veće rezolucije.

Rezolucija prognoziranih značajki	Model	Srednjeročno	
		mIoU (%)	mIoU-MO (%)
32×64 (/32)	F2F-5	51.13	46.60
	F2F-8	51.60	47.02
64×128 (/16)	F2F-5	52.41	48.16
	F2F-8	**53.40**	**49.85**

Identičan eksperiment s većom rezolucijom prognoziranih značajki isprobali smo i s F2MF modelom. Rezultati u tablici 5.6 pokazuju kako veća rezolucija značajki još više utječe na performanse F2MF modela nego li F2F-a. Kod F2MF-a zamjećujemo rast od čak 2.18 postotnih bodova mIoU, što implicira da je veći kapacitet F2MF-a dobro iskorišten za zaključivanje sa složenijim i finijim značajkama.

Problem, koji bi se mogao javiti uz značajke veće rezolucije, jest da korelacijski modul neće moći prepoznati velike pomake objekata zbog ograničenog okvira pretraživanja. Naime, okvir pretraživanja korelacijskog modula F2MF modela je kvadrat stranica veličine 9 piksela, odnosno maksimalan pomak koji se može detektirati je četiri piksela u bilo kojem smjeru. Korištenjem značajki veće rezolucije povećali smo i pomake za faktor $2\times$ te time efektivno prepolovili udaljenost za koju se objekti mogu udaljiti, a da ih korelacijski modul prepozna. Problem velikih pomaka mogli bismo riješiti povećavanjem okvira pretraživanja, no to pak dovodi do kvadratnog povećanja računske složenosti korelacijskog modula koji već ionako koristi velik udio računalnih resursa pri zaključivanju. Unatoč tome, proveli smo eksperimente i sa F2MF modelom koji koristi okvire pretraživanja veličine 11 i 13 piksela. Povećanje okvira pretraživanja na 11 piksela rezultiralo je povećanjem točnosti na 57.54% mIoU, dok model s okvirom veličine 13 piksela postiže 57.30% mIoU. Smatramo kako ova blaga regresija performansi proizlazi iz činjenice da povećanjem okvira pretraživanja također i kvadratno povećavamo broj izlaznih kanala korelacijskog modula. Tako s okvirom veličine 9 piksela dobivamo 243 kanala, 363 za okvir veličine 11, a 507 za okvir veličine 13. Na korelacije se također nadodaju i značajke sa 128 kanala pa lako možemo zaključiti da je 256 kanala premalo za prenošenje cjelokupne informacije koju predajemo dijeljenim

slojevima F2MF modela. Smatramo kako bismo bolji rezultat mogli dobiti povećanjem broja kanala u dijeljenim slojevima što bi opet negativno utjecalo na brzinu zaključivanja. Također je važno uočiti da je povećanje točnosti kod mjere mIoU-MO mnogo veće u odnosu na klasičnu mjeru mIoU, budući da ova promjena ciljano utječe upravo na značajke koje predstavljaju pomične objekte.

Tablica 5.6: Prognoziranje značajki dimenzija 32×64 i 64×128 dobivenih trorazinskim SwiftNetom s dvije i tri preskočne veze. Performanse F2MF modela znatno rastu koristeći značajke veće rezolucije. Izraz u zagradama u stupcu koji opisuje model označava veličinu okvira pretraživanja korelacijskog modula.

Rezolucija prognoziranih značajki	Model	Srednjeročno	
		mIoU (%)	mIoU-MO (%)
32×64 (/32)	F2MF (korelacije: 9)	55.03	50.97
64×128 (/16)	F2MF (korelacije: 9)	**57.21**	**54.04**
	F2MF (korelacije: 11)	**57.54**	**54.76**
	F2MF (korelacije: 13)	57.30	54.28

Temeljem eksperimenata prikazanih tablicamaTablica 5.5 iTablica 5.6 primjećujemo kako korištenjem značajki veće rezolucije uvijek dobivamo veće povećanje točnosti mjere mIoU-MO u odnosu na mjeru mIoU. Tako kod F2F-8 modela mIoU raste za 1.80 postotnih bodova, a mioU-MO za 2.83 postotna boda. Nadalje, kod F2MF modela rast mIoU je 2.18 postotnih bodova, a mioU-MO 3.07 postotnih bodova. Budući da mjera koja uzima u obzir samo pomične objekte raste otprilike 50% više u odnosu na klasičnu mIoU mjeru, možemo zaključiti da ovom modifikacijom uspijevamo bolje modelirati dinamiku objekata što je iznimno bitno za primjene poput autonomne vožnje.

Rad [9] također razmatra i fino ugađanje (eng. *fine-tuning*) prognostičkog modela i dijela segmentacijskog modela za naduzorkovanje nakon treniranja samog prognostičkog modela. Ovaj pristup isprobali smo nakon učenja modela koji rade na višoj rezoluciji značajki no naši eksperimenti nisu rezultirali poboljšanjem točnosti. Smatramo kako je to do znatno manje mini-grupe veličine samo četiri uzorka koju smo morali koristiti pri finom ugađanju zbog memorijskog ograničenja.

5.3.3. Modificiranje veličine konvolucijske jezgre

Kao što smo već spomenuli, povećavanjem rezolucije značajki efektivno otežavamo prognostičkom modelu postizanje većeg receptivnog polja te modeliranje većih pomaka objekata. Ovaj učinak pokušali smo poništiti povećavanjem konvolucijskih jezgri. Eksperimente smo, opet radi računske složenosti i broja provedenih treniranja, proveli s fokusom na manje modele poput F2F-4 i F2F-8, ali je testiran i veći F2MF model. Eksperimenti su uključivali povećanje svih jezgri deformabilnih konvolucija ili povećanje samo jezgre prvog deformabilnog sloja. Također, velik je dio eksperimenata uključivao povećanje samo konvolucijske jezgre koja se koristi za računanje pomaka, dok je sama deformabilna jezgra ostala nepromijenjene veličine. U tu svrhu, u nekim smo eksperimentima koristili i dilatirane konvolucije umjesto običnih, gustih jezgri. Polazišna motivacija iza povećanja jezgre za računanje pomaka bila je omogućiti modelu da ranije, i na većim udaljenostima, uoči objekte koji se pomiču velikom brzinom te da na temelju te informacije može bolje prilagoditi pomake deformabilne jezgre. Smatramo kako veličinu deformabilne jezgre nije potrebno mijenjati kako bismo postigli ovaj učinak. Naposljetku, ove smo modifikacije isprobali i na značajkama niže (/32) te na značajkama više (/16) rezolucije. Zbog velikog broja različitih isprobanih konfiguracija ove rezultate teško je prikazati tablično pa ćemo ih ukratko prokomentirati u sljedećih nekoliko rečenica. Većina naših eksperimenata rezultirala je blagom redukcijom točnosti ili sličnim rezultatom kao kod modela bez modificiranih jezgri. Pri prognoziranju značajki više rezolucije neke su konfiguracije postigle bolju točnost. Ovo opažanje izraženije je kod manjih modela poput F2F-4, no slično se poboljšanje može dobiti i dodavanjem još deformabilnih slojeva, što je računski efikasnije.

5.3.4. F2F prognoziranje uz korelacijski modul

Konačnu modifikaciju koju ćemo predstaviti svojevrsni je „hibrid“ između F2F i F2MF modela. Modifikacija uključuje dodavanje korelacijskog sloja F2F modelu kako bismo odredili koliko ovaj sloj utječe na poboljšanje točnosti bez ostatka F2MF modela. Korelacijski koeficijenti donose informacije važne za određivanje pomaka deformabilne jezgre što bi trebalo pozitivno utjecati na performanse modela.

Sve eksperimente proveli smo na značajkama niže rezolucije te su prikazani u tablici 5.7 Značajke korelacijskog modula dodaju velik broj kanala kada se konkateniraju na semantičke značajke te smo stoga dijeljenim deformabilnim slojevima postavili broj kanala na 256 kao i u F2MF modelu. Dodavanjem korelacija F2F-8 modelu povećali smo točnost za 1.29 postotnih bodova mjere mIoU. Taj rezultat pokazuje da velik dio poboljšanja koje F2MF donosi možemo pripisati F2M i težinskom modulu.

Iz ovoga se eksperimenta rodila ideja korištenja korelacija za težinsko miješanje u odvojenom deformabilnom sloju. Odvojeni deformabilni sloj već smo opisali na početku poglavlja, a sada bismo htjeli iskoristiti korelacije za kombiniranje značajki koje dobivamo iz ovoga sloja. Već znamo da informacija koju donosi korelacijski sloj pospješuje izračun težina u F2MF modelu pa ćemo istu ideju primijeniti na težine za kombiniranje značajki. Tako ćemo težine računati deformabilnim slojem koji na ulazu prima korelacije i značajke iz sva četiri trenutka. Ovime smo dobili dodatno poboljšanje performansi F2F-8 modela, koji s prvotnih 51.60% mIoU sada postiže 53.21% mIoU. Problem dodavanja korelacijskog modula je njegov utjecaj na brzinu zaključivanja, na što dodatno utječe povećavanje broja kanala u deformabilnim slojevima. Ovakav model ima sličnu računsku složenost kao i F2MF model, a postiže osjetno manju točnost pa nam nije previše zanimljiv.

Tablica 5.7: Metode kombiniranja značajki nakon odvojenog deformabilnog sloja. Kao osnovica usporedbe koristi se nemodificirani F2F-8, kod ostalih modela samo je prvi deformabilni sloj odvojen, dok je preostalih sedam slojeva identično kao i u F2F-8 modelu.

Model	Srednjeročno	
	mIoU (%)	mIoU-MO (%)
F2F-8	51.60	47.02
F2MF	55.03	50.97
F2F-8 + korelacijski modul	52.89	48.56
F2F-3 + korelacijski modul + odvojeni sloj	52.36	48.16
F2F-8 + korelacijski modul + odvojeni sloj	53.21	49.25

Smanjivanjem broja deformabilnih slojeva na tri možemo postići bolji balans računske složenosti i performansi. F2F-3 s navedenim modifikacijama postiže točnost od 52.36%

mIoU, a po brzini zaključivanja nalazi se otprilike na sredini između nemodificiranih F2F-8 i F2MF modela. Ovaj rezultat ilustrira kako plitki prognostički modeli mogu postići puno bolje performanse ako pametno iskoristimo korelacijske značajke, no ipak smatramo da ovaj pristup donosi nedovoljno poboljšanje performanse u usporedbi s cijenom povećanja brzine zaključivanja. Više riječi o brzini zaključivanja bit će u poglavlju 5.5, u kojima ćemo razmotriti i neke od modificiranih modela iz tablice 5.7

5.4. Učenje na većem skupu slika

Prognostičke modele do sada smo učili na neoznačenim podacima uzimajući iz svake sekvence jedan primjer za učenje. Mogućnost učenja naših modela nad neoznačenim podacima vrlo je korisno svojstvo jer je lako prikupiti još takvih sekvenca za učenje.

U sklopu ovog rada nećemo koristiti dodatne podatke u svrhu poboljšanja performansi modela kako bi naši rezultati bili usporedivi sa sličnim radovima. Međutim, tijekom dosadašnjih smo eksperimenata koristili samo jedan dio svake Cityscapes sekvence te bismo mogli povećati naš skup za učenje uzimanjem više uzoraka iz svake sekvence, slično kao u radu [1]. Odlučili smo se za četiri uzorka iz sekvence pri čemu je razmak između svakog uzorka tri sličice. Vremenski razmak od tri sličice jednak je razmaku između značajki iz četiri trenutka koje koristimo za prognoziranje te je djeljiv s razmacima za kratkoročno ($\Delta t=3$) i srednjeročno ($\Delta t=9$) prognoziranje. Opisano nam svojstvo omogućuje da iskoristimo dio već postojećih značajki za nove uzorke. Prije smo iz svake sekvence na disk zapisali sedam dobivenih mapa semantičkih značajki, a sada će biti potrebno dodati još tri mape, što ne predstavlja veliko povećanje u prostornom zauzeću diska. Dodatno, koristit ćemo nasumično horizontalno zrcaljenje kako bismo umjetno povećali skup za učenje. Budući da se uzorci mogu sastojati od prošlih značajki, ciljnih značajki te segmentacijske mape, bilo je lakše implementirati zrcaljenje prilikom samog dohvata podataka, nego napisati kompleksnu transformaciju. Zrcaljene podatke računamo za vrijeme treniranje te ih ne zapisujemo na disk radi prostornog ograničenja. Tako smo unutar naše torch.utils.data.Dataset klase, koja predstavlja skup Cityscapes, dodali zastavicu kojom odabiremo želimo li prilikom učitavanja zrcaliti podatke iz uzoraka temeljem nasumičnog odabira. Eksperimente smo proveli sa F2MF modelom prognozirajući značajke rezolucija /16 i /32 kao što je opisano u prethodnim poglavljima.

Provedeni eksperimenti prikazani su u tablici 5.8 U prvom se dijelu tablice nalaze već predstavljeni rezultati trenirani na standardnom Cityscapes skupu koji su analizirani u ranijim poglavljima. Drugi dio tablice predstavlja rezultate dobivene treniranjem na proširenom skupu uz navedenu augmentaciju podataka. F2MF na nižoj rezoluciji postiže rast točnosti od jednog postotnog boda mjere mIoU, dok mjera mIoU-MO raste nešto više. Isti model na višoj rezoluciji postiže 58.35% mIoU, što je slično poboljšanje poput onog koje zapažamo na nižoj rezoluciji, no zato postiže mnogo veći rast mjere mIoU-MO. F2MF ostvaruje točnost od 55.95% mIoU-MO što je rast od gotovo tri postotna boda u odnosu na model treniran bez dodatnih podataka. Veće poboljšanje mjere, koja u obzir uzima samo pokretne objekte, već smo uočili kod određenih modifikacija predstavljenim u prijašnjem poglavlju te zaključili kako je to dobar pokazatelj da naše promjene omogućuju bolje modeliranje kretanja objekata u sceni i bolju generalizaciju. Uz povećanje okvira pretraživanja korelacijskog modula na 11 piksela, F2MF pri rezoluciji /16 postiže 58.77% mIoU te 56.16% mIoU-MO što je ujedno i najbolji prognostički rezultat kojeg smo uspjeli dobiti.

Tablica 5.8: Srednjeročni rezultati prognoziranja F2MF na originalnom te proširenom (i augmentiranom) skupu Cityscapes. Za usporedbu su prikazani i rezultati iz rada [1] gdje za dobivanje semantičkih značajki i maske koristio veći model DenseNet-121.

Model	Srednjeročno	
	mIoU (%)	mIoU-MO (%)
F2MF /32 (DenseNet-121) [1]	56.80	53.10
F2MF /32	55.03	50.97
F2MF /16	57.21	54.04
F2MF /32 + veći skup za treniranje (DenseNet-121) [1]	57.90	54.60
F2MF /32 + veći skup za treniranje	**56.04**	**52.71**
F2MF /16 + veći skup za treniranje	**58.35**	**55.95**
F2MF /16 (korelacije: 11) + veći skup za treniranje	**58.77**	**56.16**

U tablici 5.8 također su predstavljeni i rezultati F2MF modela iz rada [1] koji je treniran kako na standardnom Cityscapes skupu, tako i na proširenom skupu, što je inspiriralo i naš

pristup. U radu [1] korištena je okosnica DenseNet za ekstrakciju značajki. Jednorazinski SwiftNet (bez preskočnih veza) uz okosnicu DenseNet-121 postiže 75.8% mIoU na skupu Cityscapes što je nešto više od piramidalnog SwiftNeta bez dijela preskočnih veza. Iz toga razloga, rezultati iz rada [1] nisu direktno usporedivi s našima, ali nam zato mogu dati uvid koliko poboljšanje možemo očekivati učenjem modela na proširenom skupu.

5.5. Brzina zaključivanja modela

Brzinu zaključivanja odabranih modela mjerili smo na grafičkoj kartici GTX 1080 Ti koja sadrži 11 GB video memorije. Istu grafičku karticu koristili smo za ubrzanje učenja i evaluacije modela u svim eksperimentima. Mjerenja smo proveli zaključivanjem uz veličinu grupe od jednog uzorka kako bismo što vjernije oponašali primjene u realnom vremenu. Prije mjerenja proveli smo „zagrijavanje“ kartice kroz 100 iteracija kako bi se svi potrebni podaci učitali u predmemoriju te frekvencija rada GPUa stabilizirala. Primjer za zaključivanje generirali smo nasumično te ga prije mjerenja učitali u video memoriju kartice kako bismo uklonili utjecaj prebacivanja podataka iz glavne memorije preko sabirnice, što može znatno usporiti zaključivanje. Zbog toga vjerujemo da su naša mjerenja brzine zaključivanja odraz računske složenosti pojedinih modela. Mjerenja smo proveli na 300 uzoraka te ćemo prezentirati srednju vrijednost vremena zaključivanja za jedan uzorak.

Tablica 5.9 prikazuje mjerenja koja smo proveli prognoziranjem značajki rezolucije 32×64 ($/32$). Značajke tih dimenzija dobivamo iz slika rezolucije 1024×2048 piksela trorazinskim piramidalnim SwiftNetom s dvije preskočne veze. Stoga prvi red tablice prikazuje brzinu zaključivanja cijelog spomenutog segmentacijskog modela, bez prognoziranja. SwiftNet postiže oko 30 fpsa (sličica u sekundi), što je prikladno za zaključivanje u realnom vremenu. Svi naši prognostički modeli na ovoj rezoluciji postižu znatno manje vrijeme odziva od odabranog segmentacijskog modela, a vrijeme zaključivanja najmanjih modela gotovo je zanemarivo u usporedbi sa SwiftNetom. Najsporiji od njih je F2MF koji zahtjeva 8.24 ms po primjeru, što je još uvijek četiri puta manje od SwiftNeta. Istaknimo, također, i F2F-3 model s korelacijskim modulom i odvojenim slojem s težinskim miješanjem značajki. On postiže kompetitivne rezultate s obzirom na to da sadrži samo tri deformabilna sloja no modifikacija ovog modela dodaje previše računske složenosti.

Tablica 5.9: Brzina zaključivanja i broj parametara prognostičkih modela pri ulaznoj rezoluciji značajki od 32×64. Značajke te rezolucije dobivamo iz trorazinskog SwiftNeta s dvije preskočne veze i okosnicom ResNet-18 pa smo u tablicu uključili i rezultate ovog segmentacijskog modela.

Model	Broj parametara (M)	Odziv (ms)	Broj sličica po sekundi (fps)
SwiftNet, 2 preskočne veze	12.05	33.69	29.68
F2F-5	0.74	1.89	529.10
F2F-8	1.25	2.85	350.88
F2MF	4.74	8.24	121.36
F2F-3 + korelacijski modul	1.47	3.38	295.86
F2F-8 + korelacijski modul	8.47	7.02	142.45
F2F-3 + korelacijski modul + težinski odvojeni sloj	1.62	4.97	201.21

U tablici 5.10 prikazane su brzine zaključivanja modela pri rezoluciji 64×128 (/16), dobivene trorazinskim piramidalnim SwiftNetom s tri preskočne veze. Prvo zanimljivo opažanje je da dodavanje još jedne preskočne veze segmentacijskom modelu zanemarivo utječe na povećanje odziva te broja parametara. Međutim, značajke više rezolucije sada sadrže četiri puta više elemenata, stoga i prognoziranje našim modelima zahtjeva gotovo točno četiri puta više vremena. Tako je brzina zaključivanja F2MF modela sada usporediva segmentacijskom modelu, dok je prognoziranje F2F modelima još uvijek primjetno brže. Tablica 5.10 također prikazuje i brzinu zaključivanja F2MF modela s većim okvirom pretraživanja (11 i 13 piksela umjesto 9) korelacijskog modula.

Tablica 5.10: Brzina zaključivanja i broj parametara prognostičkih modela pri ulaznoj rezoluciji značajki od 64×128. Značajke te rezolucije dobivamo iz trorazinskog SwiftNeta s tri preskočne veze i okosnicom ResNet-18.

Model	Broj parametara (M)	Odziv (ms)	Broj sličica po sekundi (fps)
SwiftNet, 3 preskočne veze	12.06	33.93	29.47
F2F-5	0.74	6.08	164.47
F2F-8	1.25	9.47	105.60
F2MF	4.74	29.41	34.00
F2MF (korelacije: 11)	5.04	32.42	30.84
F2MF (korelacije: 13)	5.39	35.86	27.87
F2F-3 + korelacijski modul	1.47	12.01	83.26
F2F-8 + korelacijski modul	8.47	25.59	39.08

5.6. Prognoziranje značajki generativnog modela

U ovom poglavlju razmotrit ćemo prognoziranje latentnih reprezentacija generativnog modela VQGAN. Cilj je prognozirane reprezentacije rekonstruirati u RGB sliku iz budućeg trenutka. Konfiguracija koju smo odabrali ima veličinu rječnika 16384 te ugrađuje ulaznu sliku u tenzor rezolucije /16. Radi jednostavnosti, u ovom ćemo radu koristiti predtrenirani VQGAN za zadatak rekonstrukcije slika. Postoji ograničen broj javno dostupnih naučenih modela, a mi smo se odlučili za inačicu treniranu na ImageNetu. Vjerujemo da, od dostupnih skupova na kojima je VQGAN treniran, jedino ImageNet sadrži dovoljno raznovrsnosti koja će omogućiti dobru generalizaciju na skupu urbanih cestovnih scena. Postoje i drugi skupovi s velikim brojem slika te rezolucijom sličnijom slikama iz Cityscapesa, no nismo mogli pronaći VQGAN predtreniran na njima. Slika 5.1 prikazuje kako navedeni rekonstrukcijski model generalizira na slici iz skupa Cityscapes.

Slika 5.1: Generalizacija VQGAN-a treniranog na ImageNetu. Lijevo – originalna slika, desno – slika rekonstruirana VQGAN-om.

Prije prognoziranja otkrili smo problem da preuzeta implementacije VQGAN-a ne podržava slike visoke rezolucije poput onih iz Cityscapesa. Zato smo se odlučili na poduzorkovanje slika na rezoluciju 512×1024 piksela. Za naš prvi pristup prognoziranju odlučili smo upotrijebiti F2F-8 model. Jedina potrebna modifikacija bila je promjena broja kanala u deformabilnim slojevima. Latentne reprezentacije VQGAN-a imaju 256 kanala pa smo prvom sloju, koji prima četiri para prošlih reprezentacija, postavili broj ulaznih kanala na 1024. Nadalje, odlučili smo se postaviti broj ulaznih i izlaznih kanala u ostalim slojevima na 256, umjesto standardnih 128. Prva treniranja rezultirala su potpuno sivim prognoziranim slikama na kojima nije bilo naznaka da je model išta naučio. Nakon nekoliko pokušaja i pogrešaka, došli smo do boljih rezultata smanjivanjem stope učenja. Konkretno, smanjili smo početnu stopu učenja s $5 \cdot 10^{-4}$ na $2 \cdot 10^{-4}$ te smo dodatno uveli eksponencijalno umanjivanje stope učenja s faktorom 0.94. Slika 5.2 prikazuje redom: ciljnu sliku, rekonstrukciju iz prve epohe, rekonstrukciju iz posljednje epohe. Na ovom primjeru uočavamo kako već nakon prve epohe možemo razaznati zamućene obrise s ciljne slike, dok su u posljednjoj epohi obrisi mnogo jasniji te već prepoznajemo pojedine objekte. Međutim, jasno je kako u ovom primjeru ne dobivamo niti približno kvalitetu slike kakva bi bila zadovoljavajuća. Isprobali smo razne modifikacije poput produbljivanja ili skraćivanja modela, dodavanja broja kanala te prognoziranja značajki prije i poslije kvantizacije. Svi su ovi pristupi dali vrlo slične rezultate, bez značajnih poboljšanja.

Slika 5.2: Rekonstrukcije latentnih reprezentacija prognoziranih F2F modelom. Gore lijevo – posljednja viđena slika, dolje lijevo – ciljna slika (neviđena), gore desno – rekonstrukcija iz prve epohe, dolje desno – rekonstrukcija iz posljednje epohe.

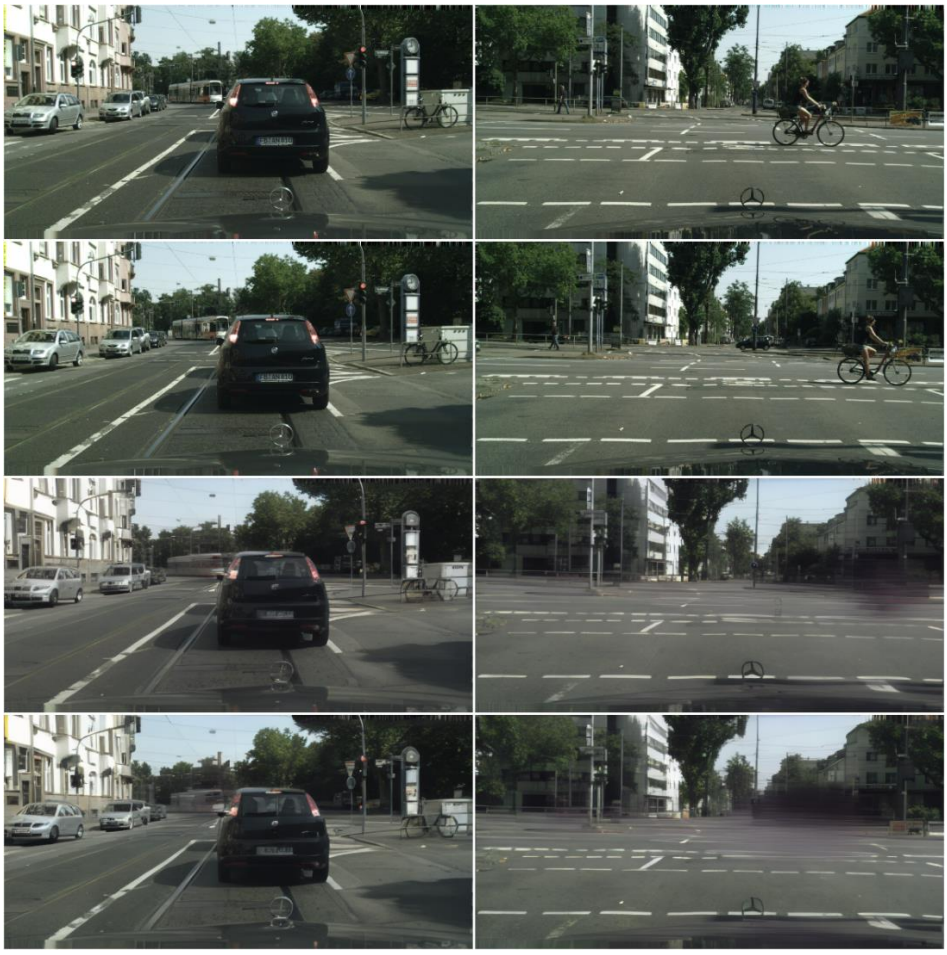
Sljedeći pristup bio je, prirodno, isprobati F2MF model na istom zadatku, no pripremili smo nekoliko promjena u odnosu na osnovnu arhitekturu. Smatramo kako isprobane prognostičke arhitekture lakše modeliraju geometrijske transformacije semantičkih nego rekonstrukcijskih značajki. Zato smo se odlučili na združeno prognoziranje obje vrste reprezentacija. Ovu arhitekturu nazvali smo multimodalni F2MF jer se na njegovom izlazu nalaze semantička i rekonstrukcijska glava. Korelacijske smo koeficijente računali temeljem semantičkih značajki radi jednostavnije implementacije i računske složenosti, no bilo bi zanimljivo ispitati mogu li se točnije korespondencije dobiti iz generativnih značajki. Semantičke i rekonstrukcijske značajke iz različitih trenutaka na ulazu konkateniramo te predajemo sloju za kombiniranje (fuzija). Takve, kombinirane, značajke i korelacijske koeficijente ponovo konkateniramo te predajemo slojevima s dijeljenim reprezentacijama koji imaju 512 kanala. Na izlazu ponovo imamo module F2M, F2F za semantičke značajke te težinski modul. Novi dodatak je i F2F modul za rekonstrukcijske značajke. Slično kao i u originalnoj arhitekturi, predviđene pomake iz F2M modula primjenjujemo zasebno i na semantičke i na rekonstrukcijske reprezentacije. Konačno, ove dvije vrste reprezentacije miješamo dijeljenim težinama koje smo predvidjeli za oba zadatka. Gubitak računamo kao zbroj srednjih kvadratnih pogreški oba izlaza (i to zasebno po modulima), slično kao i u originalnoj arhitekturi.

Latentne reprezentacije VQGAN-a dobili smo iz slika rezolucije 512×1024, no naši dosadašnji segmentacijski modeli učili su na slikama $2\times$ veće rezolucije. Kako bismo ovo uskladili, naučili smo i SwiftNet na slikama niže rezolucije. Eksperimentalno smo potvrdili kako u ovom slučaju konfiguracija s dva sloja piramide postiže nešto bolje performanse od konfiguracije s tri sloja. Dvoslojni SwiftNet s dvije preskočne veze postiže 69.38% mIoU na nižoj slikovnoj rezoluciji. Oba modela, VQGAN i SwiftNet daju značajke /16 rezolucije, no možda bismo nešto bolju točnost mogli postići sa SwiftNetom koji daju značajke /32 rezolucije, a treniran je na slikama originalne veličine.

Prognoziranje semantičkih značajki multimodalnim F2MF modelom postiže 50.36% mIoU. Ovo je nešto slabija točnost u odnosu na rezultate prikazane u prijašnjim poglavljima, no to je bilo za očekivati s obzirom na to da je segmentacijski model korišten za ekstrakciju značajki osjetno lošiji. Slika 5.3 prikazuje rekonstrukciju prognoziranih latentnih reprezentacija iz istog modela. Primijetite kako je rekonstrukcija nešto preciznija u odnosu na prognoziranje F2F modelom. U pozadini se razaznaju grane drveća, a prije nije bilo jasno koji je to objekt u pitanju. Također, prepoznajemo konture automobila u pokretu na desnoj strani slike što prije nije bilo moguće.

Slika 5.3: Rekonstrukcija latentnih reprezentacija prognoziranih multimodalnim F2MF modelom. Gore – posljednja viđena slika, sredina – ciljna slika, dolje – rekonstrukcija iz posljednje epohe.

Ovakvi rezultati mogli bi nas dovesti do zaključka kako naši prognostičke arhitekture nisu u stanju prognozirati značajke koje se mogu rekonstruirati u realistične i oku ugodne slike. Međutim, to nije sasvim točno. Naime, detaljnijim pregledom dobivenih rekonstrukcija otkrili smo da naši modeli prognoziraju ispravne reprezentacije u slučajevima kada je taj dio scene statičan u svim viđenim vremenskim trenucima. Na prikazu 5.4 (lijevo) uočavamo kako su dobro rekonstruirani svi dijelovi scene osim tramvaja koji je bio u pokretu pa je taj dio slike zamućen. Na desnoj strani istog prikaza uočavamo sličnu situaciju; jedini zamućeni dio slike je oko prognozirane lokacije biciklista u pokretu. Također primijetite kako je uz multimodalni F2MF model manji dio slike zamućen te je lokacija biciklista preciznije prognozirana u odnosu na rekonstrukciju uz F2F model. U prikazu 5.5 nailazimo na sličnu situaciju, ali su u ova dva slučaja veći dijelovi scene u pokretu pa je i rekonstrukcija zamućenija.

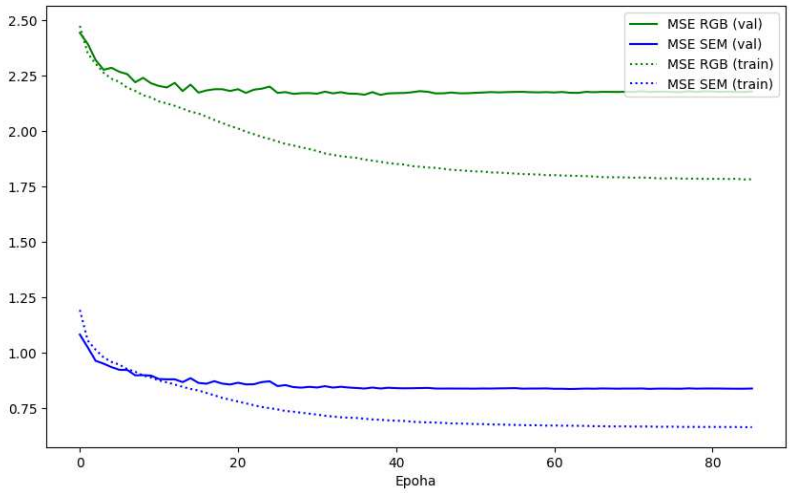


Slika 5.4: Primjeri boljih rekonstrukcija prognoziranih značajki u statičnim dijelovima scene. Slike odozgo prema dolje predstavljaju: posljednju viđenu sliku, ciljnu sliku, rekonstrukciju multimodalnim F2MF modelom, rekonstrukciju F2F modelom.

Slika 5.5: Primjeri boljih rekonstrukcija prognoziranih značajki u statičnim dijelovima scene. Slike odozgo prema dolje predstavljaju: posljednju viđenu sliku, ciljnu sliku, rekonstrukciju multimodalnim F2MF modelom, rekonstrukciju F2F modelom.

Kao provjeru da naš postupak treniranja ne rezultira prenaučenim modelima pratili smo ponašanje gubitka na skupu za treniranje i validaciju te smo vizualno usporedili prognozirane rekonstrukcije s ova dva skupa. Na slici 5.6 nalazi se graf koji prikazuje kretanje funkcija gubitka na skupu za učenje i validaciju. Gubitak srednje kvadratne pogreške računali smo zasebno za semantičke i generativne značajke, a njihov zbroj koristili smo pri učenju multimodalnog F2MF modela. Iz ovog grafa može se opaziti kako nema znakova prenaučenosti jer gubitak na validacijskom skupu ne raste značajno nakon postizanja minimuma. Pojava prenaučenosti ionako ne bi bila problem jer smo koristili rano zaustavljanje prilikom treniranja. Drugo opažanje iz grafa je da je gubitak generativnih značajki više od dva puta veći od gubitka semantičkih značajki, odnosno da je taj gubitak

prevladavao prilikom učenja. Slika 5.7 prikazuje prognozirane rekonstrukcije na skupu za učenje gdje opažamo slično ponašanje kao na validacijskom skupu.



Slika 5.6: Graf kretanja funkcija gubitka na skupu za učenje i validacijskom skupu pri treniranju multimodalnog F2MF modela. Zelenom bojom predstavljen je gubitak generativnih značajki, a plavom semantičkih značajki.

Slika 5.7: Primjeri rekonstrukcija sa skupa za učenje. Slike odozgo prema dolje predstavljaju: posljednju viđenu sliku, ciljnu sliku, rekonstrukciju multimodalnim F2MF modelom.

Zaključak

Semantičko prognoziranje predviđa semantiku na razini piksela temeljem slika iz prošlih trenutaka. Primjena ove metode posebno je zanimljiva u sustavima u kojima je potrebno anticipirati buduće položaje objekata kako bismo omogućili pravovremenu reakciju na njih. Najraširenija primjena takvih sustava danas je u autonomnim vozilima, gdje oni povećavaju sigurnost putnika i ostalih sudionika prometa, no također se mogu primijeniti i za razne druge robotske zadatke.

U ovom radu ispitali smo nekoliko modifikacija arhitektura za prognoziranje semantičkih značajki. Naši prijedlozi ciljaju povećanju točnosti proširivanjem receptivnog polja, prognoziranjem finijih značajki više rezolucije i kombiniranjem značajki iz različitih trenutaka, a osim ovih modifikacija eksperimentalno smo ispitali i mnoge manje arhitekturalne promjene. Posebno ističemo prognoziranje značajki više rezolucije, čime smo u svim modelima postigli poboljšanje točnosti veće od dva postotna boda mjere mIoU. Rast mjere mIoU-MO još je veći, u nekim slučajevima i do četiri postotna boda. Veći rast mjere mIoU-MO sugerira kako naše modifikacije omogućuju preciznije modeliranje dinamike pokretnih objekata, što je iznimno bitno za navedenu primjenu u autonomnim vozilima. Uz treniranje na proširenom skupu Cityscapes, naš F2MF model postiže vrlo kompetitivne rezultate. Također pridajemo veliku pažnju i povećanju računske složenosti koje uvode modifikacije kako bi se ovi modeli mogli koristiti u stvarnim uvjetima. Naš najsporiji model još uvijek je brži od samog modela za semantičku segmentaciju koji je napravljen za korištenje u realnom vremenu.

Osim prognoziranja semantike, razmatrali smo i korištenje modela F2F i F2MF za prognoziranje budućih RGB sličica u videu. Predstavili smo i multimodalnu arhitekturu F2MF za združeno prognoziranje semantičkih značajki i latentnih reprezentacija generativnog modela. Motivacija iza ovog pristupa bila je iskoristiti informaciju o pomacima dobivenu iz semantičkih značajki, s kojima već znamo da naši modeli dobro rade, i primijeniti je na reprezentacije generativnog modela. Ovime dobivamo određeno poboljšanje u vizualnoj kvaliteti prognoziranih slika u odnosu na direktno prognoziranje samih generativnih značajki, no dobivene rekonstrukcije još uvijek su zamućene u pokretnim dijelovima scene, a prognoziranje semantike postiže lošiju točnost. Statični dijelovi scene ne

predstavljaju problem našim prognostičkim modelima te rekonstrukcije ispadaju vizualno ugodne.

Ovaj rad ostavlja nekoliko pravaca za napredak i budući rad. Jedan od njih svakako je prognoziranje značajki iz modela za semantičku segmentaciju koji koristi jaču okosnicu, poput DenseNeta-121, te modul SPP koji ugrađuje globalne informacije u reprezentacije. Smatramo kako ove dvije nadogradnje imaju potencijal postići točnost mjerljivu s trenutnim stanjem tehnike, a zanimljivo bi bilo isprobati i kako dodavanje samog modula SPP utječe na rezultate prognoziranja. Dodatno poboljšanje moglo bi se dobiti finim ugađanjem parametara semantičkog modela, zaduženih za naduzorkovanje, s obzirom na prognoziranu semantičku mapu uz grafičku karticu s većom memorijom. Također, bilo bi zanimljivo isprobati sličan pristup finim ugađanjem dekodera VQGAN-a uz prognoziranu RGB sliku te fiksnim rječnikom ugrađivanja. Trebalo bi razmotriti i prognoziranje samih indeksa iz rječnika umjesto kvantiziranih reprezentacija ili usporediti ove rezultate s prognoziranjem reprezentacija nekog drugog generativnog modela..

Literatura

[1] Šarić, J., Oršić, M., Antunović, T., Vražić, S., Šegvić, S. *Warp to the Future: Joint Forecasting of Features and Feature Motion*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, (2020.), str. 10645–10654.

[2] Šarić, J. *Združeno prognoziranje značajki i njihova pomaka za predviđanje semantičke budućnosti u videu*. Doktorski rad. Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, 2022.

[3] Oršić, M., Šegvić, S. *Efficient semantic segmentation with pyramidal fusion*. Pattern Recognition, 110(4), 107611 (2021.)

[4] Oršić, M., Krešo, I., Bevandić, P., Šegvić, S. *In Defense of Pre-Trained ImageNet Architectures for Real-Time Semantic Segmentation of Road-Driving Images*, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, (2019.), str. 12599–12608.

[5] He, K., Zhang, X., Ren, S., Sun, J. *Deep Residual Learning for Image Recognition*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, (2016.), str. 770–778.

[6] Vandenhende, S., Georgoulis, S., Van Gansbeke, W., Proesmans, M., Dai, D., Van Gool, L. *Multi-Task Learning for Dense Prediction Tasks: A Survey*, IEEE Transactions on Pattern Analysis and Machine Intelligence vol. 44,7, (2022.), str. 3614-3633

[7] He, K., Zhang, X., Ren, S., Sun, J. *Identity Mappings in Deep Residual Networks*, Proceedings of the 14th European Conference on Computer Vision (ECCV), Amsterdam (2016.), 630–645

[8] Ioffe, S., Szegedy, C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,* Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML) - Volume 37, Lille, (2015.), str. 448–456

[9] Šarić, J., Oršić, M., Antunović, T., Vražić, S., Šegvić, S. *Single Level Feature-to-Feature Forecasting with Deformable Convolutions*, Proceedings of the 41st German Conference on Pattern Recognition (GCPR), Dortmund (2019.), str. 189-202

[10] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y. *Deformable Convolutional Networks*, IEEE International Conference on Computer Vision (ICCV), Venice, (2017.), str. 764-773

[11] Fischer, P., Dosovitskiy, A., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T. *FlowNet: Learning Optical Flow with Convolutional Networks*, Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, (2015.), str. 2758-2766

[12] Esser, P., Rombach, R., Ommer, B. *Taming Transformers for High-Resolution Image Synthesis*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, (2021.), str. 12873-12883

[13] van den Oord, A., Vinyals, O., Kavukcuoglu, K. *Neural discrete representation learning*, Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, (2017.), str. 6309–6318

Sažetak

Gusto prognoziranje predviđa reprezentacije iz budućnosti, oslanjajući se pritom na reprezentacije iz prijašnjih trenutaka. Radi se o skupu metoda, između ostalog primjenjivih u industriji autonomnih vozila te za prognoziranje sličica u videu. Ovaj rad razmatra pristupe prognoziranju semantičkih značajki, dobivenih modificiranim piramidalnim SwiftNetom, i latentnih rekonstrukcijskih reprezentacija iz VQGAN-a. Sve eksperimente provodimo na podatkovnom skupu Cityscapes. Za prognoziranje koristimo već postojeće arhitekture F2F i F2MF, a predlažemo i nekoliko modifikacija ovih modela sa svrhom poboljšanja točnosti. Pritom posebno obraćamo pažnju na utjecaj ovih preinaka na brzinu zaključivanja, odnosno računsku složenost. Od modifikacija ističemo prognoziranje značajki više rezolucije, kojim smo postigli poboljšanje točnosti oba modela za dva postotna boda mjere mIoU, dok je mjera mIoU-MO, koja uključuje samo pokretne objekte, narasla za tri postotna boda. Također, predlažemo multimodalnu arhitekturu za zajedničko prognoziranje semantičkih i rekonstrukcijskih reprezentacija. Prognoziranje rekonstrukcija VQGAN-a rezultiralo je zamućenjem slike iz budućeg trenutka u dijelovima scene koji su pokretni te ispravnim i jasnim rekonstrukcijama u statičkim dijelovima scene.

Ključne riječi: gusto prognoziranje, semantička segmentacija, rekonstrukcija slika, duboko učenje, računalni vid, piramidalna fuzija, VQGAN, Cityscapes

Summary

Dense forecasting is a method that aims to predict future representations by utilizing information from previous images. It encompasses a range of techniques that find applicability in various domains such as autonomous vehicles and video prediction. This paper explores different approaches to dense forecasting by leveraging modified pyramidal SwiftNet architecture as semantic feature extractor and utilizing VQGAN for extracting latent representations used for reconstruction. We evaluate our experiments on the Cityscapes dataset. The forecasting process utilizes existing F2F and F2MF architectures, and introduces several model modifications with the objective of enhancing prediction accuracy. Particular attention is given to assessing the impact of these modifications on inference speed and computational complexity. Notably, the inclusion of higher-resolution feature forecasting yields a two percentage-point improvement in the mIoU metric for both models, while the mIoU-MO metric, focusing exclusively on moving objects, exhibits a three percentage-point increase. Furthermore, a multimodal architecture is proposed for joint forecasting of semantic and reconstruction representations. The forecasting of VQGAN reconstructions results in blurring of in-motion sections of the scene, while achieving accurate and precise reconstructions in static sections.

Keywords: dense forecasting, semantic segmentation, image reconstruction, deep learning, computer vision, pyramidal fusion, VQGAN, Cityscapes

Skraćenice

fps	*frames per second*	broj sličica u sekundi
F2F	*feature to feature*	iz značajki u značajke
F2M	*feature to motion*	iz značajki u pomake
F2MF	*feature to motion-feature*	iz značajki u pomake i značajke
GB	*gigabyte*	gigabajt
GPU	graphics processing unit	grafička kartica
I2I	*image to image*	iz slike u sliku
M	*million*	milijun
mIoU	*mean intersection over union*	srednji omjer presjeka i unije
ms	*millisecond*	milisekunda
MSE	*mean square error*	srednja kvadratna pogreška
M2M	*motion to motion*	iz pomaka u pomak
RGB	*red, green, blue*	crvena, zelena, plava
S2S	*sematics to semantics*	iz semantičkih predikcija u semantičke predikcije