

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1548

Detekcija sigurnosnih atributa prometnica u snimkama

Ivan Relić

Zagreb, srpanj 2017.

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

Zagreb, 15. ožujka 2017.

DIPLOMSKI ZADATAK br. 1548

Pristupnik: **Ivan Relić (0036475070)**

Studij: Računarstvo

Profil: Računarska znanost

Zadatak: **Detekcija sigurnosnih atributa prometnica u snimkama**

Opis zadatka:

Standardizirani sigurnosni atributi omogućavaju objektiviziranu procjenu sigurnosti cestovnih prometnika. U trenutačnoj praksi te atributte cestovnim odsječima dodjeljuju plaćeni prometni eksperti. Glavni nedostatci takvog pristupa su cijena ljudskog rada te gubitak objektivnosti zbog subjektivne procjene eksperta. Zbog toga se javlja potreba za tehničkim sustavima koji bi dodjelu sigurnosnih atributa mogli provoditi prema standardu označavanja zadanom primjerima.

U okviru rada, potrebno je proučiti i ukratko opisati postojeće pristupe za detekciju događaja u slijedu slika. Predložiti arhitekturu dubokog modela za detekciju sigurnosnih atributa prometnika. Uhodati postupke učenja i validiranja hiperparametara. Primijeniti naučene modele na snimkama prometnika. Prikazati i ocijeniti ostvarene rezultate. Predložiti pravce budućeg razvoja.

Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Zadatak uručen pristupniku: 10. ožujka 2017.

Rok za predaju rada: 29. lipnja 2017.

Mentor:


Izv. prof. dr. sc. Siniša Šegvić

Djelovođa:


Doc. dr. sc. Tomislav Hrkač

Predsjednik odbora za
diplomski rad profila:


Prof. dr. sc. Siniša Srbljić

Zahvaljujem mentoru, izv. prof. dr. sc. Siniši Šegviću, na svim savjetima i pruženoj pomoći tijekom studiranja i tijekom izrade ovog diplomskog rada.

Zahvaljujem svojim roditeljima i sestri na pruženoj potpori tijekom studiranja.

Zahvaljujem svim svojim prijateljima koji su obaveze na fakultetu učinili lakšima, zabavnijima i koji su godine studiranja pretvorili u najljepše godine života.

SADRŽAJ

1. Uvod	1
2. Skupovi podataka i njihovo korištenje	3
2.1. Video snimke projekta FTTS iRAP	4
2.2. Skup podataka s eksperimentalnim oznakama	5
2.3. Skup podataka s oznakama iz sustava FTTS iRAP	6
3. Metode, modeli i postupci	10
3.1. Unakrsna entropija kao funkcija gubitka	11
3.2. Gradijentni spust	12
3.2.1. Učenje postupkom Adam	12
3.2.2. Regularizacija modela	14
3.3. Tipični slojevi dubokih modela za primjenu u računalnom vidu	15
3.3.1. Potpuno povezani sloj	15
3.3.2. Konvolucijski sloj	18
3.3.3. Sloj sažimanja	20
3.3.4. Sloj grupne normalizacije (engl. <i>batch normalization</i>)	21
3.4. Arhitekture korištene u eksperimentima	22
3.4.1. Arhitektura VGG-16 modela	23
3.4.2. Arhitektura za klasifikaciju pojedinačnih slika	24
3.4.3. Arhitekture za klasifikaciju sekvenci slika	25
3.5. Postupak pripreme skupova podataka	28
3.5.1. Preprocesiranje slika	29
3.5.2. Postupak kreiranja slika s oznakama iz sustava FTTS iRAP	30
4. Eksperimenti i rezultati	32
4.1. Način učenja i testiranja u eksperimentima	32
4.2. Klasifikacija ručno označenih slika na različitim rezolucijama	34

4.2.1. Rezultati na rezoluciji 700x280	35
4.2.2. Rezultati na rezoluciji 525x210	37
4.2.3. Rezultati na rezoluciji 350x140	39
4.2.4. Rezultati na rezoluciji 175x70	41
4.3. Klasifikacija pojedinačnih slika označenih konzistentno s projektom FTTS iRAP na originalnoj rezoluciji	43
4.4. Klasifikacija sekvenci slika označenih konzistentno s projektom FTTS iRAP koristeći vremensko-prostorno sažimanje	45
4.5. Klasifikacija sekvenci slika označenih konzistentno s projektom FTTS iRAP koristeći povratne LSTM ćelije	47
4.6. Klasifikacija sekvenci slika označenih konzistentno s projektom FTTS iRAP koristeći vremenski potpuno povezani sloj	49
4.7. Analiza rezultata	51
4.7.1. Analiza rezultata na ručno označenim slikama	51
4.7.2. Analiza rezultata na slikama označenima konzistentno sa sustavom FTTS iRAP	51
5. Zaključak	58
Literatura	59

1. Uvod

Računalni vid vrlo je značajno područje umjetne inteligencije koje se bavi razumijevanjem slike, odnosno scene na slici. Vrlo popularna područja računalnog vida danas su klasifikacija (određivanje pripadnosti scene slike predefiniranom skupu klase, jednoj ili više njih), semantička segmentacija (podjela slike u dijelove i dodjeljivanje klase svakom od tih dijelova), stereoskopska rekonstrukcija (određivanje 3D geometrije scene iz slika dobivenih parom kamera) te mnoga druga. Važnost percepcije i razumijevanja okoline dovela je do toga da je računalni vid vrlo popularno područje umjetne inteligencije koje je u stalnom razvoju zbog svojih mnogih primjena.

Duboko učenje donijelo je veliku revoluciju na mnogim područjima umjetne inteligencije, pa tako i na području računalnog vida. Duboki modeli pomaknuli su granice najbolje ostvarenih rezultata na mnogim problemima. Primjerice, na problemu klasifikacije slika iz skupa podataka ImageNet [7] u jednu od tisuću mogućih klasa, najmanja pogreška ostvarena dubokim modelom manja je od pogreške koju na tom skupu podataka ostvaruju ljudi [10]. Upravo zbog vrlo dobrih rezultata koje duboki modeli ostvaruju, sve više zadataka u kojima ljudi lako grijese i koji su mentalno naporni povjeravaju se na obavljanje računalima.

Dobri rezultati dubokih modela na području računalnog vida svakako otvaraju mnoge primjene računala u praktičnim problemima. Jedan takav problem, kojim se ovaj rad bavi, jest inspekcija kvaliteta cesta. Organizacija iRAP (engl. *International Road Assessment Programe*) ima za cilj upravo inspekciju kvaliteta ceste pridruživanjem različitih sigurnosnih atributa segmentima prometnica. Zadatak se obavlja tako da se na temelju video snimaka prometnica uz geolokacije, pridružene video snimkama, određuje prisutnost sigurnosnog atributa na tom dijelu prometnice. Na temelju prisutnih atributa moguće je procijeniti sigurnost promatranog dijela prometnice, što je vrlo korisno.

Trenutno, pridruživanje atributa segmentima prometnica obavljaju ljudi. Zadatak je vrlo naporan i skup te su moguće različite greške – greške uzrokovane subjektivnošću, greške uzrokovane radom različitih osoba na zadatku pridruživanja zbog čega

se može pojaviti nekonzistentnost, greške uzrokovane umorom te mnoge druge. Prednost korištenja dubokog modela za taj zadatak je njegova veća brzina, manja cijena i objektivnost u procjeni u odnosu na čovjeka.

Ovaj rad kroz nekoliko faza opisuje implementaciju dubokog modela koji je sposoban detektirati prisutnost sigurnosnog atributa u video snimci. Prvi dio rada opisuje korišteni skup podataka – postupak njegovog prikupljanja te korištenje za učenje dubokog modela. Drugi dio rada opisuje korištene metode i modele koji na temelju učenja na skupu podataka ostvaruju željenu funkciju koju možemo primijeniti na konkretni problem. Završni dio rada opisuje provedene eksperimente i dobivene rezultate prilikom korištenja dubokog modela za problem detekcije sigurnosnih atributa u snimkama.

2. Skupovi podataka i njihovo korištenje

Prepoznavanje prisutnosti sigurnosnih atributa možemo okarakterizirati kao binarnu klasifikaciju čiji konačni izlaz nam govori je li atribut prisutan u sceni ili nije. Standard organizacije iRAP propisuje 78 sigurnosnih atributa [4] koje je moguće pridružiti segmentima prometnica. Neki od njih su: tip križanja, ograničenje brzine, udaljenost i tip objekata pored ceste, broj prometnih traka i tako dalje. Pojedine atribute moguće je detektirati u statičnim slikama, dok je neke moguće detektirati samo iz video snimke – primjerice, da bismo odredili ograničenje brzine nekog segmenta prometnice, potrebna nam je video snimka cijelog tog segmenta.

Modeli dubokog učenja danas uspješno rješavaju probleme klasifikacije. Rješavanje takvog problema temelji se na odabiru prikladnog modela uz dovoljan broj označenih podataka. Ključna stvar prilikom rješavanja problema klasifikacije jest izvlačenje znanja iz označenih podataka koje model koristi. Takav postupak nazivamo nadziranim učenjem [9]. Učenje modela svodi se na korekciju slobodnih parametara modela koji ostvaruju željenu funkciju preslikavanja.

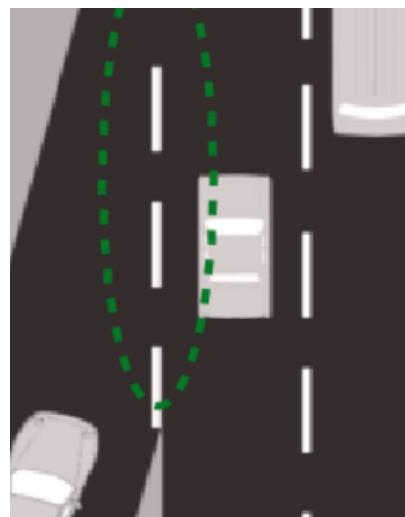
Skup podataka možemo podijeliti na 3 podskupa koji imaju različite uloge:

- podskup za učenje – podskup koji se koristi za korekciju slobodnih parametara modela
- podskup za validaciju – podskup koji se koristi za korekciju hiperparametara, odnosno za provjeru sposobnosti generalizacije modela
- podskup za testiranje – podskup koji se koristi za konačno ispitivanje performanse naučenog modela

2.1. Video snimke projekta FTTS iRAP

Kao osnova za kreiranje skupova podataka, korištena je stranica sustava FTTS iRAP [1]. Stranica nudi sučelje u kojem je moguće pregledavati, uređivati i prikupljati sigurnosne atribute prisutne na segmentima prometnica. Uz same atribute dodijeljene segmentima prometnica, dostupne su i georeferencirane video snimke – moguć je pregled video snimke koja geolokacijom odgovara segmentu. Video snimke snimljene su iz automobila koji se kretao po engleskim autocestama te ih je moguće, uz pripadajuće informacije o njihovim geolokacijama, slobodno preuzeti sa stranice [1].

Sustav opisan u radu koristi georeferencirane video snimke s navedene stranice za učenje. Konkretno, kako je riječ o binarnoj klasifikaciji, potrebni su nam označeni podaci. Atribut koji sustav prepozna je pripajanje (engl. *merge lane*) (tip križanja). Kako je riječ o video snimkama s engleskih cesta, pripajanja koja se prepoznaju bit će s lijeve strane. Definicija atributa pripajanje prema [3] jest sljedeća – promet koji prilazi sa strane pripaja se preko linije za pripajanje na cestu kojoj se atribut dodjeljuje.



Slika 2.1: Simbolički prikaz pripajanja (slika preuzeta i prerađena iz [3])

Slika 2.1 simbolički prikazuje što se konkretno smatra pripajanjem prilikom pridjeljivanja sigurnosnog atributa pripajanja. Prilikom označavanja, eksperti za označavanje kao pripajanje označavaju onaj dio prometnice u kojem se automobil iz traka prometnice koja se pripaja mora prebaciti u trak prometnice kojoj se pripaja. Kao pripajanje se promatra upravo taj dio prometnice jer predstavlja opasnost zbog toga što vozač prilikom ubacivanja mora promijeniti trak kojim prometuje.

Iz georeferenciranih video snimaka kreirana su dva tipa skupova podataka:

- skup podataka s eksperimentalnim oznakama (oznake atributa nisu konzistentne

- s pravilima koja propisuje organizacija iRAP)
- skup podataka s oznakama iz sustava FTTS iRAP (oznake atributa su konzistentne s pravilima koja propisuje organizacija iRAP)

U nastavku je opisan način kreiranja i uporaba pojedinih tipova skupova podataka.

2.2. Skup podataka s eksperimentalnim oznakama

Početna ideja bila je, neovisno o pridruženim atributima pripajanja sa stranice [1], označiti pripajanja eksperimentalnim oznakama.

Dubokim modelima je, kao i ljudima, lakše izlučiti znanje iz podataka ukoliko u njima postoje diskriminativne značajke koje mogu odrediti pripadnost određenoj klasi. Dodjeljivanje oznaka prisutnosti atributa pripajanja prometnoj sceni iz tog se razloga vodilo tom mišlju – scene na kojima su prisutne karakteristične, diskriminativne značajke koje bi mogle odrediti prisutnost atributa pripajanja označene su kao pozitivne, odnosno scene na kojima je atribut prisutan, dok su ostale označene kao negativne, odnosno scene na kojima atribut nije prisutan.



Slika 2.2: Prikaz scene na kojoj je prisutna diskriminativna značajka atributa pripajanja (slika preuzeta s [1])

Slika 2.2 prikazuje eksperimentalno označenu pozitivnu sliku. Na slici su vidljive karakteristične bijele trake u uzorku koji bi potencijalno mogao biti diskriminativan prilikom donošenja odluke modela o prisutnosti atributa pripajanja.



Slika 2.3: Prikaz scene na kojoj diskriminativna značajka atributa pripajanja nije prisutna (slika preuzeta s [1])

Slika 2.3 prikazuje eksperimentalno označenu negativnu sliku. Vidljiv je izostanak karakterističnih bijelih traka na cijeloj sceni.

Skup podataka s eksperimentalnim oznakama dobiven je obradom 7 video snimaka preuzetih sa stranice [1]. Ukupan broj dobivenih označenih slika s raspodjelom u podskupove je sljedeći:

- podskup za učenje – 1796 slika, od toga 898 pozitivnih
- podskup za validaciju – 626 slika, od toga 313 pozitivnih
- podskup za testiranje – 594 slike, od toga 297 pozitivnih

Skup je kreiran na način da se iz svake video snimke uzorkuju pozitivni primjeri (primjeri na kojima je prisutna diskriminativna značajka atributa pripajanja) te se zatim uzorkuje isti broj negativnih primjera na kojima diskriminativna značajka karakterističnog bijelog uzorka nije prisutna.

2.3. Skup podataka s oznakama iz sustava FTTS iRAP

U skupu podataka s eksperimentalnim oznakama, pozitivne i negativne oznake određene su na temelju prisutnosti diskriminativnog uzorka u sceni. Takav način označavanja nije konzistentan s načinom označavanja kakav provode eksperti za označavanje u sustavu FTTS iRAP na stranici [1]. Glavni cilj rada jest razviti sustav koji bi mogao automatski određivati prisutnost sigurnosnog atributa na prometnoj sceni. Za učenje je stoga potrebno kreirati skup podataka koji je označen konzistentno s pravilima koja propisuje iRAP, odnosno koji će koristiti oznake iz sustava FTTS iRAP.

U dogovoru s kreatorima stranice [1], izvučeni su podaci geolokacija svih segmenta prometnica na kojima je prisutan atribut pripajanja. Segmenti na kojima se određuje je li sigurnosni atribut prisutan ili nije dugački su 10m. Također, moguće

je i otvoriti video snimku koja prikazuje trenutak kada započinje segment s atributom pripajanja.



Slika 2.4: Prikaz satelitske snimke s označenim segmentom atributa pripajanja (slika preuzeta s [2])



Slika 2.5: Prikaz video snimke s označenim segmentom atributa pripajanja sa slike 2.4 (slika preuzeta s [1])



Slika 2.6: Prikaz satelitske snimke s označenim segmentom atributa pripajanja (slika preuzeta s [2])



Slika 2.7: Prikaz video snimke s označenim segmentom atributa pripajanja sa slike 2.6 (slika preuzeta s [1])



Slika 2.8: Prikaz satelitske snimke s označenim segmentom atributa pripajanja (slika preuzeta s [2])



Slika 2.9: Prikaz video snimke s označenim segmentom atributa pripajanja sa slike 2.8 (slika preuzeta s [1])

Promotrimo slike 2.4, 2.6 i 2.8. Na slikama je crvenom linijom označen segment koji je na stranici [1] prema pravilima koje propisuje iRAP označen kao pripajanje. Već sa satelitske snimke je vidljivo kako bi moglo biti problema prilikom detekcije tog atributa zbog nedostatka diskriminativnih značajki koje bi to mogle olakšati. Slike 2.5, 2.7 i 2.9 redom prikazuju isječke iz video snimaka u trenucima kada započinju segmenti sa slika 2.4, 2.6 i 2.8 označeni kao pripajanja. Iz same slike je vrlo teško odrediti da je u tom trenutku prisutan atribut pripajanja.

Problem nedostatka informacije možemo riješiti tako da odluku ne temeljimo na jednoj slici nego na sekvenci slika koje su prethodile trenutnoj. Koristeći sekvencu slika koja prethodi slici označenoj kao početak segmenta označenog kao pripajanje, model će biti u mogućnosti lakše donijeti odluku zbog prisutnosti diskriminativne značajke (karakteristične bijele trake u uzorku). Prisutnost diskriminativne značajke u sekvenci koja prethodi početku pripajanja vidljiva je i iz satelitske snimke. Skup podataka s oznakama iz sustava FTTS iRAP iz tog razloga ne sadrži statične, pojedinačne slike nego sekvence slika.

Skup podataka s oznakama iz sustava FTTS iRAP dobiven je obradom 100 videa preuzetih sa stranice [1]. Ukupan broj dobivenih označenih sekvenci s raspodjelom u podskupove je sljedeći:

- podskup za učenje – 7554 sekvenci, od toga 3777 pozitivnih
- podskup za validaciju – 1720 sekvenci, od toga 860 pozitivnih
- podskup za testiranje – 1642 sekvenci, od toga 821 pozitivnih

Skup je kreiran na način da se iz svake video snimke uzorkuju pozitivni primjeri na sljedeći način:

- odrede se rasponi sličica video snimke koji su prema stranici [1] označeni kao pozitivni (sadrže atribut pripajanja)
- za svaku sličicu označenu kao pozitivnu uzorkuje se i nekoliko sličica unatrag koje će zajedno činiti jednu pozitivnu sekvencu

Na sličan način uzorkuje se i jednak broj negativnih primjera za one sličice za koje je prema stranici [1] označeno da ne sadrže atribut pripajanja. Cjelokupni postupak kreiranja skupa podataka uz konkretnе podatke o duljini sekvence bit će dan u nastavku rada.

3. Metode, modeli i postupci

Sustav koji ovaj rad opisuje temelji se na algoritmima dubokog učenja. Zadatak koji je potrebno riješiti jest binarna klasifikacija, odnosno određivanje je li atribut pripajanja prisutan u prometnoj sceni. Model koji rješava zadatak tada možemo prikazati sljedećom funkcijom:

$$\mathbf{y} = f^*(\mathbf{x}; \boldsymbol{\phi}), \quad (3.1)$$

gdje je f^* funkcija koju želimo naučiti na podacima. Argumenti funkcije f^* su podatak \mathbf{x} (slika ili sekvenca slika – isječak video snimke) te skup slobodnih parametara, $\boldsymbol{\phi}$. Vrijednost funkcije f^* , odnosno \mathbf{y} predstavlja upravo željenu informaciju koja nas zanima – vjerojatnost da je atribut pripajanja prisutan, odnosno vjerojatnost da atribut pripajanja nije prisutan na prometnoj sceni koja odgovara ulaznom vektoru podataka, \mathbf{x} .

Modeli dubokog učenja prepostavljaju kompozitnu strukturu ulaznih podataka – npr. čovjek je građen od glave i tijela, glava je građena od očiju, usta, nosa i ušiju, oči su građene od rožnice, zjenice i šarenice, i tako dalje. Iz tog razloga se nad ulaznim podacima obavljaju višestruke linearne i nelinearne operacije [21]. Funkciju f^* stoga možemo prikazati i na sljedeći način: $f^*(\mathbf{x}; \boldsymbol{\phi}) = f^{(n)}(f^{(n-1)}(\dots f^{(2)}(f^{(1)}(\mathbf{x}, \boldsymbol{\phi}), \boldsymbol{\phi}), \boldsymbol{\phi}), \boldsymbol{\phi})$ [9]. Pojedine funkcije $f^{(i)}$ predstavljaju slojeve dubokog modela koje obavljaju različite operacije.

Ocjenu prikladnosti funkcije f^* sa slobodnim parametrima $\boldsymbol{\phi}$ možemo mjeriti funkcijom gubitka. Funkcija gubitka nam govori koliko dobro naša funkcija sa slobodnim parametrima aproksimira stvarna preslikavanja koja su definirana na skupu podataka za učenje.

Učenje modela, odnosno optimizacija, tada se svodi na pretraživanje za onim skupom slobodnih parametara koji će minimizirati funkciju gubitka nad skupom podataka za učenje. Optimizacijski postupak koji se koristi je gradijentni spust nad funkcijom gubitka koju deriviramo po slobodnim parametrima modela.

Nastavak rada opisuje korištenu funkciju gubitka, optimizacijsku metodu za uče-

nje modela te tipične slojeve dubokih modela koji se koriste za primjenu u računalnom vidu. Opisane su i konkretne arhitekture koje su se koristile prilikom provođenja eksperimenata te detaljni postupci kojima su generirani skupovi podataka.

3.1. Unakrsna entropija kao funkcija gubitka

Unakrsna entropija kao funkcija gubitka definira se kao negativni logaritam izglednosti modela nad podacima. Učenjem želimo maksimizirati izglednost parametara nad podacima za učenje, odnosno želimo parametre modela za koje je najizglednije da minimiziraju razliku između funkcije f koju aproksimiramo i aproksimirane funkcije modela f^* .

Prepostavke za izvod unakrsne entropije kao funkcije gubitka su sljedeće:

- zadatak koji rješavamo jest zadatak klasifikacije slike u jednu od k različitih klasa
- izlazni vektor je dimenzionalnosti k i označava međusobno zavisne vjerojatnosti da ulazni podatak pripada svakoj od k klasa
- podaci za učenje su izvučeni iz iste razdiobe i međusobno su nezavisni

Izvedimo unakrsnu entropiju kao funkciju gubitka iz logaritma izglednosti modela nad podacima prema [22]:

$$\begin{aligned}
 C &= -\ln \mathcal{L}(\phi, \mathcal{D}) = -\ln(p_{model}(\mathcal{D}|\phi)) \\
 &= -\ln \prod_{\mathbf{x} \in \mathcal{D}} \prod_{j=1}^k p_{model}(\mathbf{y}_j | \mathbf{x}, \phi)^{\mathbf{y}_j} \\
 &= -\ln \prod_{\mathbf{x} \in \mathcal{D}} \prod_{j=1}^k f_j^*(\mathbf{x})^{\mathbf{y}_j} \\
 &= -\sum_{\mathbf{x} \in \mathcal{D}} \sum_{j=1}^k \ln(f_j^*(\mathbf{x})) \mathbf{y}_j
 \end{aligned} \tag{3.2}$$

gdje je k dimenzionalnost izlaznog vektora, \mathbf{y}_j je stvarni izlaz (0 ili 1), a $f_j^*(\mathbf{x})$ je vrijednost j -te komponente izlaznog vektora za primjer \mathbf{x} iz skupa podataka za učenje. Vjerojatnost $p_{model}(\mathbf{y}_j | \mathbf{x}, \phi)$ predstavlja vjerojatnost oznake primjera. Kako su izlazi modela iz intervala $[0, 1]$, navedenu vjerojatnost možemo zapisati kao $f_j^*(\mathbf{x})^{\mathbf{y}_j}$.

3.2. Gradijentni spust

Gradijentni spust je iterativni optimizacijski algoritam za minimiziranje vrijednosti funkcije. U općenitom slučaju, izračun nove vrijednosti argumenta ϕ funkcije y čija se vrijednost minimizira, definiran je kao:

$$\phi_{n+1} = \phi_n - \psi \frac{\partial y}{\partial \phi}, \quad (3.3)$$

gdje je ψ pozitivna, uobičajeno mala konstanta koja kontrolira faktor korekcije parametra.

U slučaju učenja dubokog modela, izraz 3.3 možemo zapisati kao:

$$\phi_{n+1} = \phi_n - \eta \frac{\partial C}{\partial \phi}, \quad (3.4)$$

gdje je ϕ skup slobodnih parametara modela, η je parametar algoritma učenja koji nazivamo stopa učenja (engl. *learning rate*) kojim definiramo koliko će se po iznosu mijenjati parametri prilikom učenja. Funkcija koju optimiramo je funkcija gubitka, C .

Postoje različite varijante osnovnog algoritma za korekciju parametara modela. Korekciju parametara moguće je vršiti akumulirajući gradijent parametara po svim primjerima iz skupa za učenje, gdje govorimo o stvarnom gradijentu nad funkcijom gubitka, C . Akumulirani gradijent se zatim koristi za korekciju parametara prema izrazu 3.4. U tom slučaju govorimo o klasičnom gradijentnom spustu.

Varijanta u kojoj se gradijent izračunat u svakom primjeru odmah koristi za korekciju parametara naziva se stohastički gradijentni spust. Stohastičnost varijante očituje se u tome što ustvari ne računamo stvarni gradijent funkcije gubitka C , nego funkcije C_x koja definira gubitak nad jednim određenim primjerom x iz skupa za učenje. Stohastički gradijentni spust pokazuje bolju otpornost na zaglavljivanje u lokalnim optimumima, ali i brže učenje zbog češće korekcije parametara.

Varijanta u kojoj se gradijent akumulira nad određenim brojem primjera i zatim se koristi za korekciju parametara naziva se grupni (engl. *mini-batch*) gradijentni spust [15].

3.2.1. Učenje postupkom Adam

Učenje postupkom Adam je varijanta postupka učenja klasičnim algoritmom gradijentnog spusta. Za potrebe učenja modela korištenog u ovom radu korišteno je učenje postupkom Adam zbog toga što je postupak robusniji i daje bolje rezultate u odnosu na klasični algoritam gradijentnog spusta [12].

Učenje postupkom Adam, za razliku od učenja klasičnim algoritmom gradijentnog spusta, kombinira dvije napredne tehnike prilikom korekcije parametara. Algoritam kombinira metode momenta i adaptivnog pomaka što pospješuje učenje dubokih modela.

Metoda momenta pomaže ubrzanju učenja na način da prilikom korekcije parametara modela u obzir uzima i iznos prethodne korekcije — samo ime i analogija dolaze iz područja fizike pa metodu možemo poistovjetiti s tromostom tijela. Korekcija parametara se, u odnosu na izraz 3.4, definira na sljedeći način:

$$\mathbf{v}_{n+1} = \alpha \mathbf{v}_n - \eta \frac{\partial C}{\partial \phi} \quad (3.5)$$

$$\phi_{n+1} = \phi_n + \mathbf{v}_{n+1} \quad (3.6)$$

Parametar α određuje jačinu momenta, odnosno utjecaj prethodne korekcije na trenutnu korekciju. Osim ubrzanja učenja, moment pospješuje i otpornost na zaglavljivanje u lokalnim optimumima jer je moguće “preskočiti” lokalni optimum u koji bismo inače zaglavili slijepo prateći samo iznos gradijenta [9] [15].

Metoda adaptivnog pomaka prepostavlja da je stabilnost gradijenta ovisna o svakoj komponenti parametara koji se optimiraju zbog toga što promjena svake pojedine komponente parametara može drugačije utjecati na promjenu iznosa funkcije gubitka koja se optimira. Iz tog razloga je smisleno koristiti zasebne stope učenja za svaku od njih i automatski im mijenjati vrijednost kroz postupak učenja, ovisno o tome kako se pojedina komponenta gradijenta mijenja. Konačan efekt jest da će se komponente parametara čije su parcijalne derivacije po funkciji gubitka vrlo visokih vrijednosti korigirati sa značajno manjim stopama učenja, a komponente parametara čije su parcijalne derivacije po funkciji gubitka relativno niskih vrijednosti, korigirat će se većim stopama učenja [9] [15]. Efekt se postiže skaliranjem vrijednosti gradijenata akumuliranim eksponencijalnim pomicnim prosjekom (engl. *exponential moving average*) što je vidljivo u pseudokodu algoritma 1. Takvim skaliranjem će gradijenti koji su veći po iznosu biti skalirani većim vrijednostima čime će se njihov konačan iznos biti kontroliran.

Algoritam 1 Učenje postupkom Adam

inicijalne vrijednosti:

$$t_0 = 0$$

$$\mathbf{m}_0 = \vec{0}$$

$$\mathbf{v}_0 = \vec{0}$$

osvježavanje parametara

$$\mathbf{g}_t = \frac{\partial C}{\partial \phi_t}$$

$$\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1) \mathbf{g}_t$$

$$\mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2) \mathbf{g}_t^2$$

$$\hat{\mathbf{m}}_t = \frac{\mathbf{m}_t}{1 - \beta_1^t}$$

$$\hat{\mathbf{v}}_t = \frac{\mathbf{v}_t}{1 - \beta_2^t}$$

$$\phi_{t+1} = \phi_t - \eta \frac{\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon}$$

Pseudokod postupka Adam koristi nekoliko hiperparametara: t je vremenski korak koji se inkrementira sa svakom korekcijom parametara, η je stopa učenja, β_1 i β_2 su faktori korišteni za eksponencijalni pomični prosjek (engl. *exponential moving average*). Preporučene vrijednosti su $\eta = 0.001$, $\beta_1 = 0.9$ i $\beta_2 = 0.999$ [12]. Parametar ϵ služi za izbjegavanje dijeljenja s nulom i postavlja se na vrlo malu vrijednost (npr. 10^{-8}).

3.2.2. Regularizacija modela

Slojevi u dubokim modelima često imaju vrlo velik broj parametara što može dovesti do problema prenaučenosti. Vrlo česta regularizacijska metoda koja nastoji spriječiti prenaučenost jest regularizacija metodom smanjenja normi parametara (engl. *weight decay*). Regularizacija metodom smanjenja normi parametara mijenja funkciju gubitka koja se optimira tako da uvodi član ovisan o normi parametara, i to na sljedeći način:

$$C^* = C + \lambda p(\phi), \quad (3.7)$$

gdje je C^* nova funkcija gubitka koju optimiramo, C je originalna funkcija gubitka, $p(\phi)$ predstavlja funkciju norme nad parametrima ϕ , a λ je regularizacijski faktor. Faktor λ određuje utjecaj regularizacije – što je on veći, regularizacija je jača i dobivamo jednostavnije modele koji nisu skloni prenaučenosti.

Motivacija za korištenje regularizacije metodom smanjenja normi parametara leži u tome da će funkcije modela koje imaju niže vrijednosti normi parametara biti zaglađenije, odnosno imat će nižu varijancu što je poželjno za bolju generalizaciju. Prisjetimo se kako funkciju f^* dubokog modela možemo prikazati kroz višeslojnu strukturu više različitih funkcija kao što je to objašnjeno u poglavlju 3. Parametri koje regulariziramo metodom smanjenja normi su oni parametri koji su u bilo kojem od slojeva u direktnoj vezi s ulaznim argumentima u funkciju tog sloja. Razlog tomu je što želimo postići da male promjene vrijednosti na ulazu u bilo koji sloj ne uzrokuju velike promjene vrijednosti na izlazu tog sloja.

Često korištene funkcije norme $p(\phi)$ za vektor slobodnih parametara dimenzionalnosti n jesu:

L1 norma $p(\phi) = \sum_{i=1}^n |\phi_i|$ — L1 (engl. *Lasso*) regularizacija

L2 norma $p(\phi) = \sqrt{\sum_{i=1}^n \phi_i^2}$ — L2 (engl. *Ridge*) regularizacija

Regularizacija L2 normom parametara je najčešće korištena zbog lakšeg izračuna izraza parcijalne derivacije za korak gradijentnog spusta.

3.3. Tipični slojevi dubokih modela za primjenu u računalnom vidu

Duboki modeli koji se koriste za primjenu u računalnom vidu tipično su građeni od nekoliko različitih slojeva koji imaju različite uloge. Slojevi obavljaju različite operacije nad podacima i tako ostvaruju željenu konačnu funkcionalnost koja je potrebna da se obavi neki zadatak.

3.3.1. Potpuno povezani sloj

Operacija koju obavlja potpuno povezani sloj definirana je na sljedeći način:

$$\mathbf{y} = g(\mathbf{W}\mathbf{x}), \quad (3.8)$$

gdje je \mathbf{x} vektor ulaznih podataka, g je aktivacijska funkcija, a \mathbf{y} je vektor izlaznih podataka. Vektor ulaznih podataka obično se proširuje na oblik $\mathbf{x} = [1 \ x_1 \ x_2 \ \dots \ x_n]^T$. Vrijednost x_1 se fiksno postavlja na vrijednost 1 iz razloga što želimo dozvoliti da model modelira pomak parametrima prvog stupca matrice \mathbf{W} . Primjetimo kako će upravo dimenzionalnost matrice \mathbf{W} određivati dimenzionalnost vektora izlaznih podataka.

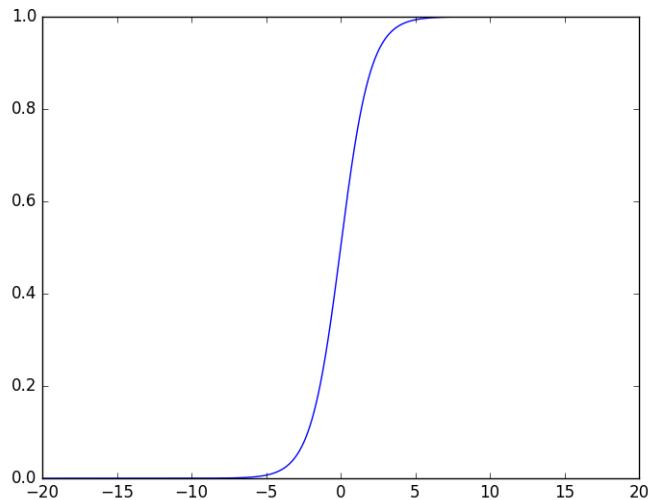
Aktivacijska funkcija g djeluje po elementima vektora dobivenog operacijom matričnog množenja matrice \mathbf{W} i vektora ulaznih podataka \mathbf{x} . Tipično se koriste nelinearne aktivacijske funkcije kako bismo dobili veću ekspresivnost dubokog modela. Postoje različite aktivacijske funkcije koje se u praksi koriste. U nastavku su opisane neke od njih.

Sigmoidalna aktivacijska funkcija

Sigmoidalna aktivacijska funkcija i njena derivacija definirane su kao:

$$g_j^{(i)}(\mathbf{x}) = \frac{1}{1 + e^{-\alpha x_j}}, \quad (3.9)$$

$$\frac{dg^{(i)}(\mathbf{x})}{d\mathbf{x}} = \alpha g^{(i)}(\mathbf{x})(1 - g^{(i)}(\mathbf{x})) \quad (3.10)$$



Slika 3.1: Graf sigmoidalne aktivacijske funkcije

Sigmoidalna aktivacijska funkcija preslikava ulaznu vrijednost na vrijednost iz intervala $\langle 0, 1 \rangle$. Sigmoidalna aktivacijska funkcija je derivabilna na cijeloj domeni i njenim korištenjem dobivamo nelinearne modele. Slika 3.1 prikazuje graf sigmoidalne aktivacijske funkcije. Iz njega je vidljivo kako će gradijent za većinu vrijednosti sigmoidalne funkcije biti vrlo niska zbog toga što je vrijednost sigmoidalne funkcije gotovo na cijeloj domeni u zasićenju (engl. *saturating function*) — asimptotski se približava vrijednosti 0 ili 1. Kako je ranije navedeno, učenje modela obavljamo koristeći gradijentni spust funkcije gubitka po parametrima modela. Uglavnom nizak iznos gradijenta

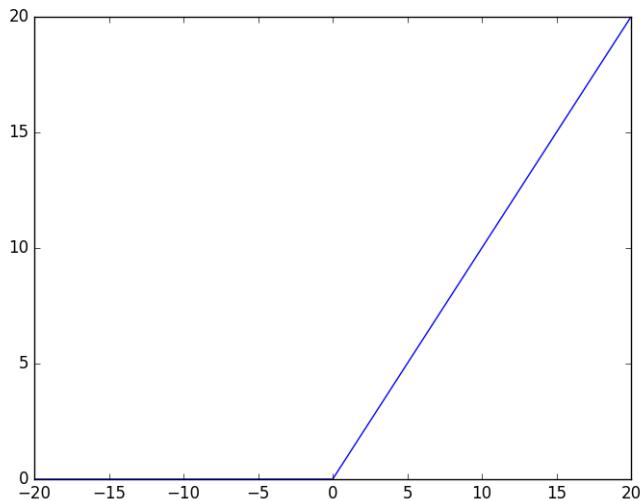
utječe na to da će već u zadnjem, izlaznom sloju korekcije parametara modela biti vrlo male. Kako izračun korekcija propagiramo unatrag, slojevi koji su sve dalje od završnog sloja imat će sve manje iznose korekcija što će uvelike utjecati na brzinu i kvalitetu učenja koje će se drastično smanjiti. Sigmoidalne aktivacijske funkcije se u praksi zato ne koriste u dubokim modelima [15].

Po dijelovima linearna aktivacijska funkcija

Po dijelovima linearna aktivacijska funkcija, ReLU (engl. *Rectified Linear Unit*) i njena derivacija definirane su kao:

$$g_j^{(i)}(\mathbf{x}) = \begin{cases} 0, & \text{ako je } x_j < 0 \\ x_j, & \text{ako je } x_j \geq 0 \end{cases}, \quad (3.11)$$

$$\frac{dg_j^{(i)}(\mathbf{x})}{d\mathbf{x}} = \begin{cases} 0, & \text{ako je } x_j < 0 \\ 1, & \text{ako je } x_j \geq 0 \end{cases} \quad (3.12)$$



Slika 3.2: Graf po dijelovima linearne aktivacijske funkcija

Po dijelovima linearna aktivacijska funkcija predstavlja funkciju praga na vrijednosti 0 ispod koje izlazne vrijednosti nisu propuštene. Po dijelovima linearna aktivacijska funkcija je često korištena u dubokim modelima zbog zanimljivih svojstava koja pokazuju prilikom učenja takvih struktura. Vrijednost gradijenta je kod po dijelovima linearne aktivacijske funkcije konstantna za polovicu cijele njene domene. U drugoj polovici iznosi 0, što ponekad također može biti problem jer može dovesti do mrtvih

dijelova slojeva kojima se parametri nikako ne korigiraju. Problem mrtvih dijelova slojeva može se riješiti korištenjem tzv. leaky ReLU prijenosne funkcije koja dozvoljava mali iznos gradijenta kada je vrijednost aktivacije 0 [9] [15].

Normalizirajuća eksponencijalna aktivacijska funkcija

Normalizirajuća eksponencijalna aktivacijska funkcija (engl. *softmax*) i njena derivacija definirane su kao:

$$g_j^{(i)}(\mathbf{x}) = \frac{e^{x_j}}{\sum_{k=1}^n e^{x_k}}, \quad (3.13)$$

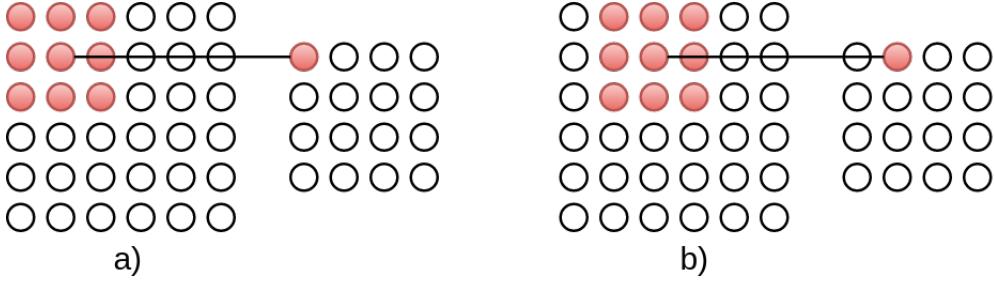
$$\frac{dg_j^{(i)}(\mathbf{x})}{\mathbf{x}} = \begin{cases} g_j^{(i)}(\mathbf{x})(1 - g_j^{(i)}(\mathbf{x})), & \text{ako je } j = k \\ -g_j^{(i)}(\mathbf{x})g_k^{(i)}(\mathbf{x}), & \text{ako je } j \neq k \end{cases} \quad (3.14)$$

Normalizirajuća eksponencijalna aktivacijska funkcija je generalizacija sigmoidalne funkcije u smislu da proširuje sigmoidalnu funkciju namijenjenu binarnoj klasifikaciji na klasifikaciju s n klasa.

Normalizirajuća eksponencijalna aktivacijska funkcija preslikava n - dimenzionalni vektor realnih vrijednosti u n - dimenzionalni vektor realnih vrijednosti koje se zbrajaju u vrijednost 1. Takav vektor realnih vrijednosti možemo promatrati kao vjerojatnosnu razdiobu i upravo iz tog razloga se normalizirajuća eksponencijalna aktivacijska funkcija često koristi u posljednjem sloju dubokih modela za probleme klasifikacije – daje nam vjerojatnost s kojom ulazni uzorak pripada svakoj od n mogućih klasa, slično kao što je sigmoidalna aktivacijska funkcija korištena kod binarne klasifikacije logističkom regresijom [15].

3.3.2. Konvolucijski sloj

Konvolucijski slojevi koriste se za stvaranje mapa značajki uporabom dvodimenzionalne konvolucije jezgre s podacima. Za definiciju konvolucije potrebno je prvo definirati veličinu jezgre (širinu i visinu), vrijednosti parametara jezgre te pomak jezgre po širini odnosno po visini prilikom izračuna.



Slika 3.3: Dva koraka konvolucije s dvodimenzionalnim podacima

Slika 3.3 prikazuje dva koraka konvolucije podataka s jezgrom. Slika a) prikazuje prvi korak konvolucije, a slika b) drugi korak. U oba koraka veličina jezgre je 3×3 , a korak iznosi 1. Korak konvolucije (engl. *stride*) određuje pomak prozora po širini, odnosno visini ulazne mape značajki.

Općenito, vrijednost jednog elementa mape značajki u sloju l dobivene konvolucijom nad jednom ulaznom mapom značajki iz sloja $l - 1$ definiramo kao:

$$x_{i,j}^l = \sum_{m,n=0}^{m=K_w, n=K_h} x_{i \cdot s_w + m, j \cdot s_h + n}^{l-1} w_{m,n}^l, \quad (3.15)$$

gdje je $x_{i,j}^l$ vrijednost mape značajki u sloju l , za redak i i stupac j . K_w je širina jezgre, K_h je visina jezgre, a $w_{m,n}^l$ vrijednost parametara jezgre za redak m i stupac n . Vrijednosti s_w i s_h predstavljaju pomak po širini, odnosno visini.

Iraz 3.15 opisuje izračun vrijednosti jednog elementa mape značajke koji je linearna kombinacija vrijednosti samo jedne mape značajki prethodnog sloja. U konvolucijskim arhitekturama se općenito koriste mape značajki koje su povezane s većim brojem mapa značajki iz prethodnog sloja. Za svaku od mapa značajki s kojom je povezana postoji po jedna jezgra koja se koristi za izračun vrijednosti. Primjerice, jedna mapa značajki ulaznog sloja može biti povezana s 3 mapa značajki - crvenom, plavom i zelenom komponentom slike u boji. Za svaku od komponenti boja će mapa značajki tada imati drugčiju jezgru.

Nadalje, izračun vrijednosti mapa značajki se često generalizira kao i kod potpuno povezanog sloja pa se dodaje dodatan parametar kao vrijednost praga te se koriste aktivacijske funkcije. Generalizirani izraz za izračun vrijednosti jednog elementa mape značajki u sloju n dobivene konvolucijom odgovarajućih jezgri sa svakom od mapa značajki iz sloja $n - 1$ s kojima je povezana definiramo kao:

$$x_{k',i,j}^l = \sigma \left(\sum_{k \in F^{l-1}} \sum_{m,n=0}^{m=K_w^k, n=K_h^k} x_{k,i \cdot s_w + m, j \cdot s_h + n}^{l-1} w_{k,m,n}^l + b^l \right), \quad (3.16)$$

gdje je $x_{k,i,j}^l$ vrijednost mape značajki k u sloju l , za redak i i stupac j . F^{l-1} je skup mapa značajki iz sloja $l-1$ s kojima je mapa značajki iz sloja l povezana. K_w^k je širina jezgre, a K_h^k je visina jezgre za mapu značajki k iz sloja $l-1$. $w_{k,m,n}^l$ je vrijednost parametara jezgre za mapu značajki k iz sloja $l-1$ za redak m i stupac n . Vrijednosti s_w i s_h predstavljaju pomak po širini, odnosno visini. Vrijednost b^l predstavlja vrijednost praga za mapu značajku iz sloja l . U pravilu se vrijednost praga dijeli za cijelu mapu značajki, no to nije nužno te je moguće za svaku od mapa značajki iz prethodnog sloja imati po jednu vrijednost praga. Konačno, funkcija σ predstavlja aktivacijsku funkciju.

Zbog svojih svojstava, konvolucijski slojevi mogu uvelike poboljšati učinkovitost modela za klasifikaciju slika. Konvolucijski slojevi su tipično rijetko povezani — nije nužno da su sve mape značajki sloja $l-1$ povezane sa svakom od mapa značajki iz sloja l [9], što uz jezgre koje su tipično mnogo manjih veličina od ulaza nad kojima djeluju u velikoj mjeri smanjuje broj parametara i povećava efikasnost učenja i računanja. Potpuno povezani slojevi koriste matrično množenje i svaki parametar je iskorišten samo jednom za opisivanje veze između ulaza i izlaza, dok se konvolucijom isti parametri koriste za više lokacija ulaza i predstavljaju svojevrsne značajke koje jezgra izlučuje na slici. Konvolucijske arhitekture korištenjem jezgara s dijeljenim parametrima uče skup parametara koji je otporan na translacije jer će za iste ulazne vrijednosti jezgra dati iste izlazne vrijednosti, bez obzira gdje na slici se one pojave. Naučeni dijeljeni skup parametara predstavlja jednu funkciju koju jezgra obavlja (primjerice, detekcija rubova) – učenjem se skup parametara prilagođava cijeloj slici i uči se tako da obavlja apstraktne funkcije neovisne o lokaciji slike, a koje će pripomoći u samoj klasifikaciji. Nabrojana svojstva konvolucijske arhitekture donose svojevrsnu ugrađenu regularizaciju što povlači bolju generalizaciju uz veću efikasnost [9] [19] [15].

3.3.3. Sloj sažimanja

Slojevi sažimanja koriste se da se grupira određeni broj podataka mape značajki i predstavi ih kao jedan podatak koji statistički dobro predstavlja tu grupu. Sam postupak također smanjuje veličinu mapa značajki i to tako da ih skalira faktorom $\frac{1}{S_w}$ po širini,

odnosno faktorom $\frac{1}{S_h}$ po visini, gdje je S_w širina, a S_h visina podmape značajki koja se grupira.

Korisnost postupka očituje se u vidu ostvarivanja invarijantnosti na lokalne translacije što može biti vrlo korisno u slučaju da nam je bitnije da je neka značajka prisutna od toga na kojoj je točno lokaciji [15].

Postoje različite vrste funkcija koje se koriste za sažimanje od kojih je u ovom radu korištena sljedeća:

sažimanje maksimalnom vrijednošću Sažimanje maksimalnom vrijednošću (engl. *max-pooling*) sažima značajke tako da iz podmape značajki nad kojom djeluje izluči samo maksimalnu vrijednost koja postaje predstavnik. Pokazuje se da se ovom metodom dobivaju vrlo dobri rezultati [17].

3.3.4. Sloj grupne normalizacije (engl. *batch normalization*)

Učenje dubokih modela može biti vrlo komplikirano ukoliko bolje promotrimo njihovu arhitekturu, način učenja i sam utjecaj načina učenja na brzinu učenja. Učenjem mijenjamo parametre modela, a samim time, promjenom parametara u određenom sloju modela, mijenjamo distribuciju ulaznih podataka koje dobiva idući sloj. Sama dubina dodatno uvećava utjecaj promjena parametara modela – distribucija ulaza svakog sloja uvjetovana je promjenama na parametre modela svih slojeva koji mu prethode.

Konstantna promjena distribucije ulaza slojeva predstavlja problem jer je potrebno stalno se prilagođavati na novu distribuciju, što može uvelike usporiti učenje. Opisani problem nazivamo interni kovarijacijski pomak (engl. *internal covariate shift*). Metoda grupne normalizacije (engl. *Batch Normalization*) uklanja taj problem i uvelike ubrzava učenje dubokih modela.

Uklanjanje problema internog kovarijacijskog pomaka rješavamo normalizacijom ulaza svakog sloja – linearno ga transformiramo tako da mu je srednja vrijednost jednaka 0, a varijanca mu je jednaka 1. Direktna normalizacija ulaza i izlaza nekog sloja ovisi o statistici svih ulaza tog sloja nad kompletним skupom podataka nad kojim učimo model. Zbog izrazite računske složenosti koja bi bila potrebna za to, uvode se neka pojednostavljenja.

Prva prepostavka za pojednostavljenje složenosti jest da umjesto normalizacije ulaza i izlaza združeno, normaliziramo svaku komponentu ulaza zasebno, s prepostavkom da komponente nisu međusobno zavisne. Normalizacija k -te komponente ulaznog vektora \mathbf{x} nekog sloja definirana je na sljedeći način:

$$\hat{\mathbf{x}}_k = \frac{\mathbf{x}_k - \mathbb{E}[\mathbf{x}_k]}{\sqrt{\text{Var}[\mathbf{x}_k]}}, \quad (3.17)$$

gdje su očekivanje i varijanca izračunati nad skupom podataka za učenje. Normalizacija svakog ulaza nekog sloja može promijeniti njegovu ekspresivnu snagu pa iz tog razloga uvodimo transformaciju normaliziranog ulaza kojom osiguravamo da je iz

normaliziranih ulaznih podataka moguće dobiti originalne vrijednosti. Transformacija k -te komponente prethodno normaliziranog ulaznog vektora $\hat{\mathbf{x}}$ nekog sloja definirana je na sljedeći način:

$$\mathbf{y}_k = \gamma_k \hat{\mathbf{x}}_k + \beta_k, \quad (3.18)$$

gdje su γ_k i β_k parametri koji se uče zajedno s parametrima modela i koji omogućuju skaliranje i pomak normalizirane vrijednosti izlaza.

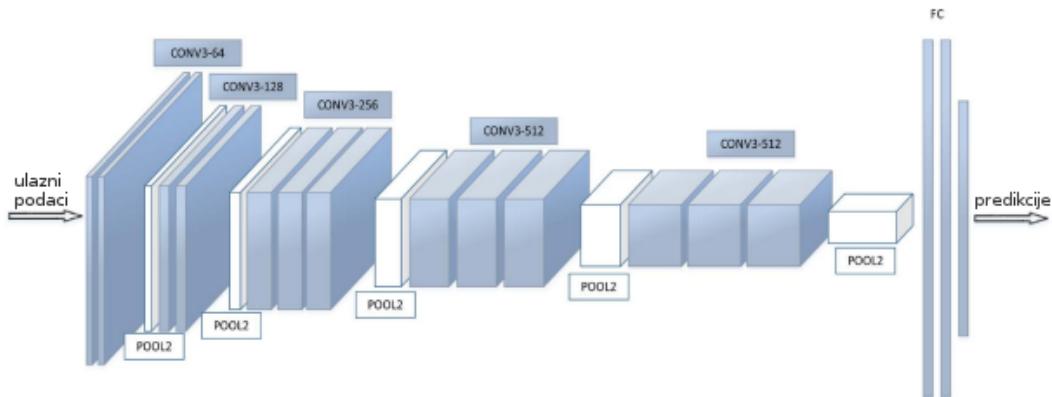
Druga pretpostavka za pojednostavljenje složenosti jest da normalizaciju umjesto nad statistikom cjelokupnog skupa podataka za učenje, obavljamo nad statistikama malih podskupova podataka (engl. *mini-batch*). Grupna normalizacija nad statistikom malog podskupa skupa podataka za učenje u nekom sloju tada je definirana kao normalizacija i linearna transformacija ulaza, definirane u izrazima 3.17 i 3.18.

Transformacija grupne normalizacije najčešće se koristi nad izračunatim težinskim sumama slojeva, prije djelovanja aktivacijske funkcije. Normalizirane vrijednosti $\hat{\mathbf{x}}_k$ možemo promatrati kao ulaze u podsloj kojem je funkcija linearna, i to je upravo funkcija transformacije definirana u izrazu 3.18. Na izlaz podsloja tada djeluje procesiranje originalnog sloja. Takvom arhitekturom ostvarili smo da svi podslojevi na ulaze dobivaju podatke s fiksnom distribucijom (s očekivanjem 0 i varijancom 1), što ubrzava učenje podslojeva, a time posljedično i samih slojeva i kompletног modela [11] [15].

3.4. Arhitekture korištene u eksperimentima

U poglavlju 2 opisana su dva tipa skupova podataka korištenih za eksperimente – skup podataka sa statičnim slikama i skup podataka sa sekvencama slika. Korišteno je nekoliko različitih modela koji su zasnovani na arhitekturi VGG-16 [18] treniranoj za natjecanje ILSVRC-2014 [16]. Model VGG-16 treniran je za klasifikaciju slika u 1000 različitih klasa na vrlo velikom broju slika – 1.3 milijuna [18]. Pokazuje se da model iz tog razloga vrlo dobro generalizira i na drugim problemima [18] pa je iz tog razloga odabran kao baza modela korištenih za eksperimente u ovom radu.

3.4.1. Arhitektura VGG-16 modela



Slika 3.4: Arhitektura VGG-16 modela, preuzeto i prerađeno iz [8]

Slika 3.4 prikazuje originalnu arhitekturu VGG-16 građenu od 16 slojeva.

Prednji dio sastoji se od 5 konvolucijskih blokova. Svaki od blokova izgrađen je na sličan način – nekoliko konvolucijskih slojeva iza kojih slijedi sloj za sažimanje maksimalnom vrijednošću (engl. *max pool*). Konvolucijski slojevi imaju jezgre dimenzija 3×3 uz izlazni korak konvolucije (engl. *stride*) 1. Svi konvolucijski slojevi koriste po dijelovima linearnu funkciju (engl. *ReLU*) te su regularizirani L2 normom parametara s faktorom regularizacije iznosa $5 \cdot 10^{-4}$. Slojevi za sažimanje maksimalnom vrijednošću imaju podmape za grupiranje veličine 2×2 uz korak uzorkovanja podmapa 2. Konvolucijski blokovi građeni su kako slijedi:

- prvi konvolucijski blok:
 - konvolucijski sloj sa 64 izlaznih mapa značajki
 - konvolucijski sloj sa 64 izlaznih mapa značajki
 - sloj za sažimanje maksimalnom vrijednošću
- drugi konvolucijski blok:
 - konvolucijski sloj sa 128 izlaznih mapa značajki
 - konvolucijski sloj sa 128 izlaznih mapa značajki
 - sloj za sažimanje maksimalnom vrijednošću
- treći konvolucijski blok:
 - konvolucijski sloj s 256 izlaznih mapa značajki
 - konvolucijski sloj s 256 izlaznih mapa značajki
 - konvolucijski sloj s 256 izlaznih mapa značajki

- sloj za sažimanje maksimalnom vrijednošću
- četvrti konvolucijski blok:
 - konvolucijski sloj s 512 izlaznih mapa značajki
 - konvolucijski sloj s 512 izlaznih mapa značajki
 - konvolucijski sloj s 512 izlaznih mapa značajki
 - sloj za sažimanje maksimalnom vrijednošću
- peti konvolucijski blok:
 - konvolucijski sloj s 512 izlaznih mapa značajki
 - konvolucijski sloj s 512 izlaznih mapa značajki
 - konvolucijski sloj s 512 izlaznih mapa značajki
 - sloj za sažimanje maksimalnom vrijednošću

Stražnji dio građen je od 3 potpuno povezana sloja koji služe za klasifikaciju konvolucijskih reprezentacija dimenzija $7 \times 7 \times 512$ u 1000 različitih klasa. Prilikom korištenja arhitekture VGG-16 za druge namjene, obično se uklanja stražnji ili svi potpuno povezani slojevi te se na prednji dio koji služi za izlučivanje značajki iz ulaznih slika nadodaju novi slojevi koji se treniraju za obavljanje specifičnog zadatka. Ovakva primjena prednaučenih konvolucijskih modela naziva se prijenos znanja (engl. *knowledge transfer*) [14].

3.4.2. Arhitektura za klasifikaciju pojedinačnih slika

Duboki model korišten za klasifikaciju pojedinačnih slika temeljen je na VGG-16 arhitekturi. Koristi se naučen prednji dio s 5 konvolucijskih blokova. Dobivene izlazne mape značajki pretvaraju se u jednodimenzionalan vektor koji se zatim vodi na potpuno povezane slojeve koji su definirani kako slijedi:

- potpuno povezani sloj dimenzionalnosti 200
 - po dijelovima linearna (engl. *ReLU*) aktivacijska funkcija
 - grupna normalizacija (engl. *batch normalization*)
 - regularizacija L2 normom parametara
- potpuno povezani sloj dimenzionalnosti 2 za binarnu klasifikaciju – određivanje je li atribut prisutan na slici ili nije
 - normalizirajuća eksponencijalna (engl. *softmax*) aktivacijska funkcija
 - regularizacija L2 normom parametara

3.4.3. Arhitekture za klasifikaciju sekvenci slika

Za klasifikaciju sekvenci slika korišteno je nekoliko različitih modela temeljenih na prednaučenoj arhitekturi VGG-16. Korišteni modeli su inspirirani arhitekturama iz [13].

Zajednički postupak koji svi modeli dijele jest postupak izlučivanja prostornih značajki – za svaku sliku sekvence se koristeći prednji dio prednaučene arhitekture VGG-16 izračunaju značajke koje se zatim dalje obrađuju na različite načine. Dobivene prostorne značajke svake pojedine slike su dimenzija $H \times W \times 512$, gdje dimenzije H i W ovise o dimenzijama ulaznih slika, a treća dimenzija je određena brojem izlaznih mapa značajki posljednjeg konvolucijskog sloja arhitekture VGG-16.

Klasifikacija sekvenci slika korištenjem vremensko-prostornog sažimanja

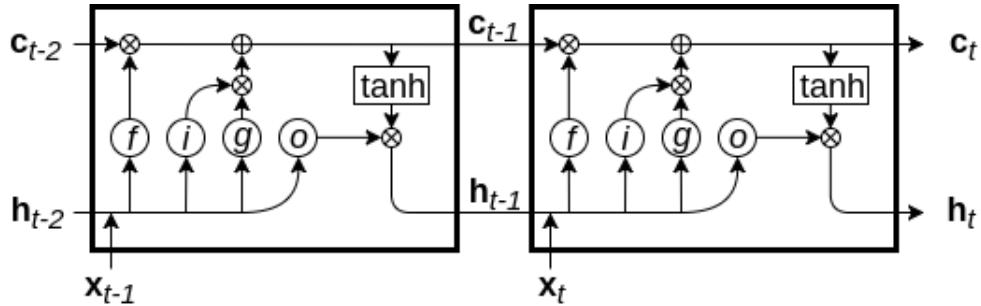
Model koji koristi vremensko-prostorno sažimanje pretvara dobivene prostorne značajke svake slike u jednodimenzionalni vektor dimenzija $H \times W \times 512$. Dobiveni jednodimenzionalni vektori se zatim oblikuju u dvodimenzionalni tenzor dimenzija $N \times (H \times W \times 512)$, gdje je N duljina sekvence slika.

Dvodimenzionalni tenzor koja se dobije povezivanjem prostornih značajki sadrži i vremenske i prostorne informacije. Taj se tenzor zatim obrađuje kroz nekoliko slojeva kako slijedi:

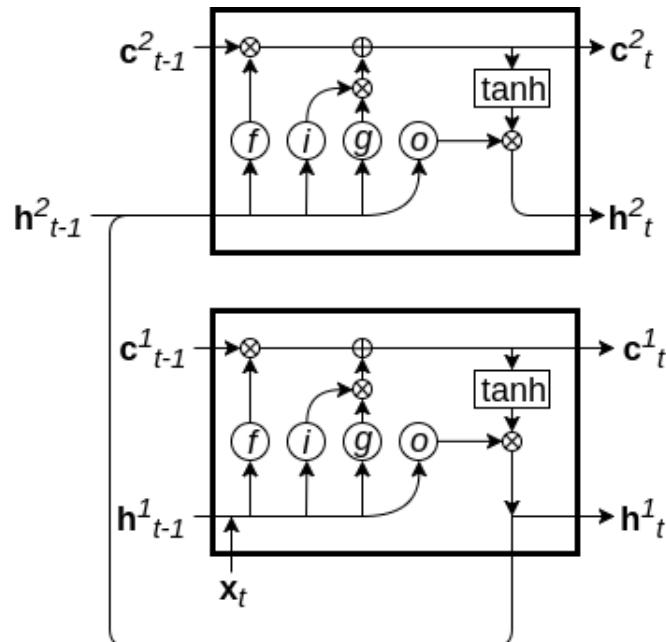
- sloj za sažimanje maksimalnom vrijednošću s podmapama za grupiranje veličina 2×2 uz korak uzorkovanja podmapa 2
 - operacija efektivno provodi vremensko-prostorno sažimanje značajki koje su bliske vremenski i prostorno
- pretvaranje dobivenog dvodimenzionalnog tenzora u jednodimenzionalni
- potpuno povezani sloj dimenzionalnosti 200
 - po dijelovima linearna (engl. *ReLU*) aktivacijska funkcija
 - grupna normalizacija (engl. *batch normalization*)
 - regularizacija L2 normom parametara
- potpuno povezani sloj dimenzionalnosti 2 za binarnu klasifikaciju – određivanje je li atribut prisutan na slici ili nije
 - normalizirajuća eksponencijalna (engl. *softmax*) aktivacijska funkcija
 - regularizacija L2 normom parametara

Klasifikacija sekvenci slika korištenjem povratnih LSTM čelija (engl. *Long Short Term Memory*)

LSTM čelije pogodne su za korištenje nad sekvencama podataka iz razloga što sadrže memoriske čelije u kojima mogu čuvati informacije kroz dugačke vremenske periode.



Slika 3.5: Arhitektura jednoslojne LSTM čelije (slika nacrtana prema [20])



Slika 3.6: Arhitektura s više slojeva LSTM čelija (slika nacrtana prema [20])

Slika 3.5 prikazuje arhitekturu jednoslojne LSTM čelije koja kao ulaz dobiva ulazni podatak. Slika 3.5 prikazuje arhitekturu s više slojeva LSTM čelija u kojoj je izlaz LSTM čelije prethodnog sloja ulaz LSTM čeliji idućeg sloja.

U svakom vremenskom koraku potrebno je izračunati izlaz LSTM čelije i novi sadržaj memoriske čelije. Izlaz LSTM čelije i novi sadržaj memoriske čelije računaju se na temelju tri vektora:

- ulaznom podatku trenutnog vremenskog koraka, odnosno izlazu LSTM ćelije prethodnog sloja trenutnog vremenskog koraka – \mathbf{x}_t na slici 3.5, odnosno \mathbf{h}_t^1 na slici 3.6
- izlazu LSTM ćelije trenutnog sloja za prethodni vremenski korak – \mathbf{h}_{t-1} na slici 3.5, odnosno \mathbf{h}_{t-1}^1 i \mathbf{h}_{t-1}^2 na slici 3.6
- sadržaju memorijске ćelije trenutnog sloja za prethodni vremenski korak – \mathbf{c}_{t-1} na slici 3.5, odnosno \mathbf{c}_{t-1}^1 i \mathbf{c}_{t-1}^2 na slici 3.6

U općenitom slučaju višeslojne arhitekture LSTM ćelija, izlaz i novi sadržaj memorij-ske ćelije računaju se sljedećim izrazima:

$$\text{stanje ćelije: } \mathbf{c}_t^l = \mathbf{f} \odot \mathbf{c}_{t-1}^l + \mathbf{i} \odot \mathbf{g} \quad (3.19)$$

$$\text{izlaz ćelije: } \mathbf{h}_t^l = \mathbf{o} \odot \tanh(\mathbf{c}_t^l) \quad (3.20)$$

$$\text{vektor ulaznih vrata: } \mathbf{i} = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}^l, \mathbf{h}_t^{l-1}] + \mathbf{b}_i) \quad (3.21)$$

$$\text{vektor vrata zaboravljanja: } \mathbf{f} = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}^l, \mathbf{h}_t^{l-1}] + \mathbf{b}_f) \quad (3.22)$$

$$\text{vektor izlaznih vrata: } \mathbf{o} = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}^l, \mathbf{h}_t^{l-1}] + \mathbf{b}_o) \quad (3.23)$$

$$\text{vektor ulaznih modulacijskih vrata: } \mathbf{g} = \sigma(\mathbf{W}_g[\mathbf{h}_{t-1}^l, \mathbf{h}_t^{l-1}] + \mathbf{b}_g), \quad (3.24)$$

gdje je tanh aktivacijska funkcija tangens hiperbolni, σ je sigmoidalna aktivacijska funkcija, \odot je operator množenja koji djeluje po elementima, a $[\mathbf{x}, \mathbf{y}]$ je operacija spajanja vektora \mathbf{x} dimenzija n_1 i vektora \mathbf{y} dimenzija n_2 u vektor dimenzija $n_1 + n_2$. Notacija \mathbf{h}_t^l predstavlja izlazni vektor LSTM ćelije vremenskog trenutka t u sloju l .

Model koji koristi LSTM ćelije prostorne značajke svake slike pretvori u jednodimenzionalni vektor dimenzija $H \times W \times 512$. Dobiveni jednodimenzionalni vektori se zatim redom obrađuju kako slijedi:

- LSTM sloj sa stanjem dimenzionalnosti 128
- LSTM sloj sa stanjem dimenzionalnosti 64
- LSTM sloj sa stanjem dimenzionalnosti 32
- potpuno povezani sloj dimenzionalnosti 2 za binarnu klasifikaciju – određivanje je li atribut prisutan na slici ili nije
 - ulaz u ovaj potpuno povezani sloj jest posljednji izlaz dobiven iz LSTM ćelije prethodnog sloja koja je obradila cijelu sekvencu podataka (vektor dimenzionalnosti 32)
 - normalizirajuća eksponencijalna (engl. *softmax*) aktivacijska funkcija
 - regularizacija L2 normom parametara

Klasifikacija sekvenci slika korištenjem vremenskog potpuno povezanog sloja

Model koji koristi vremenski potpuno povezani sloj prostorne značajke pretvori u jednodimenzionalni vektor dimenzija $H \times W \times 512$. Svaki dobiveni jednodimenzionalni vektor prostornih značajki se zatim obrađuje kako slijedi:

- potpuno povezani sloj dimenzionalnosti 64
 - po dijelovima linearna (engl. *ReLU*) aktivacijska funkcija
 - grupna normalizacija (engl. *batch normalization*)
 - regularizacija L2 normom parametara
 - ovaj potpuno povezani sloj služi za predstavljanje prostornih značajki vektorom manjih dimenzija

Jednodimenzionalni vektori dobiveni nakon potpuno povezanog sloja se zatim oblikuju u dvodimenzionalni tenzor dimenzija $N \times 64$, gdje je N duljina sekvence slika. Taj se tenzor zatim obrađuje kroz nekoliko slojeva kako slijedi:

- pretvaranje dobivenog dvodimenzionalnog tenzora u jednodimenzionalni
- potpuno povezani sloj dimenzionalnosti 64
 - po dijelovima linearna (engl. *ReLU*) aktivacijska funkcija
 - grupna normalizacija (engl. *batch normalization*)
 - regularizacija L2 normom parametara
 - ovaj potpuno povezani sloj služi za povezivanje dobivenih prostornih značajki po dimenziji vremena
- potpuno povezani sloj dimenzionalnosti 2 za binarnu klasifikaciju – određivanje je li atribut prisutan na slici ili nije
 - normalizirajuća eksponencijalna (engl. *softmax*) aktivacijska funkcija
 - regularizacija L2 normom parametara

3.5. Postupak pripreme skupova podataka

Poglavlje 2 opisuje dva tipa skupova podataka korištenih u ovom radu. U nastavku je opisan postupak pretprocesiranja slika koje se dobivaju iz georeferenciranih video snimaka sa stranice [1] te postupak kreiranja skupa podataka s oznakama iz sustava FTTS iRAP.

3.5.1. Pretprocesiranje slika

Slike korištene za kreiranje skupova podataka dobivene su iz video snimaka snimljenih automobilom koji se kretao engleskim autocestama. Zbog kretnje automobila uzorkovane slike bit će snimljene iz različitog kuta u odnosu na Sunce. Takva situacija za posljedicu ima da su neke slike mnogo svjetlijе, a neke mnogo tamnije od drugih.

Metoda kojom se može ublažiti velika varijacija kontrasta između slika jest metoda adaptivnog izjednačavanja histograma [6]. Izjednačavanje histograma za cilj ima izjednačiti razdiobu intenziteta svih kanala slike (crvenog, zelenog i plavog) na način da oni budu uniformno distribuirani. Metoda adaptivnog izjednačavanja histograma za računanje histograma ne uzima u obzir cijelu sliku nego prozore određene veličine čime se dobiva na lokalnosti izjednačavanja.



Slika 3.7: Originalna slika (slika preuzeta s [1])



Slika 3.8: Slika s izjednačenim histogramom



Slika 3.9: Slika s adaptivno izjednačenim histogramom

Iz slika 3.7, 3.8 i 3.9 vidljivo je kako su, u odnosu na originalnu sliku, na slici s adaptivno izjednačenim histogramom najbolje naglašeni detalji zbog najbolje ublažene varijacije kontrasta. Eksperimentalno je utvrđeno da adaptivno izjednavačanje histograma poboljšava performanse dubokog modela. Metoda adaptivnog izjednačavanja histograma se iz tog razloga koristi kao korak pretprocesiranja za svaku sliku korištenu prilikom kreiranja skupova podataka.

3.5.2. Postupak kreiranja slika s oznakama iz sustava FTTS iRAP

Kreiranje slika prometnih scena s oznakama iz sustava FTTS iRAP obavlja se automatiziranim postupkom. Ulazni podaci za automatizirani postupak su sljedeći:

- geolokacije na kojima je prisutan atribut pripajanja
- popis poveznica na sve video snimke u sustavu [1]
- broj video snimaka koji će se obraditi

Postupak će iz ulaznih podataka kreirati sljedeće:

- slika s oznakom prisutnosti atributa pripajanja na njoj te sekvenca slika koja joj prethodi (kao što je objašnjeno u poglavlju 2)
- slika s oznakom prisutnosti atributa pripajanja bez sekvenca koja joj prethodi
- geolokacije označenih slika i sekvenci slika

Postupak kreira označene slike sa i bez sekvence koja joj prethodi kako bi se eksperimentima ustanovilo dobiva li se na točnosti postupka uvodeći više informacija kroz sekvencu slika.

Geolokacije video snimaka uzorkovane su rijetko – najčešće svakih 0.5s. Iz tog razloga se za točnu geolokaciju konkretne sličice video snimke vrši linearna interpolacija između dvije uzorkovane geolokacije koje su vremenski najbliže konkretnoj sličici. Geolokacija konkretnih sličica korisna je informacija iz razloga što je prilikom

testiranja modela za svaki ulazni podatak kojem model dodijeli krivu oznaku moguće u sustavu FTTS iRAP provjeriti je li osoba koja je dodijelila vrijednost atributa tom segmentu to napravila točno ili je riječ o mogućoj pogrešci. Takvim postupkom moguće je na brži način validirati ispravnost dodijeljenih oznaka u sustavu.

Nastavak opisuje sažeti pseudokod automatiziranog postupka:

Algoritam 2 Automatizirano kreiranje slika označenih konzistentno s projektom FTTS iRAP

Sve dok nije obrađen željen broj video snimaka **Ponavljam**

```
v=preuzmi_video()
geolok,vrem=preuzmi_geolokacije_i_vremena(v)
sl=kreiraj_sličice(v)
ind_geolok,brz_aut=indeksiraj_geolokacije_i_izračunaj_brzine_automobila(geolok,vrem)
ras_poz=izračunaj_raspone_pozitiva(geolok_pripajanja,ind_geolok,brz_aut)
poz_slič,geolok_slič=kreiraj_positivne_sličice_s_geolokacijama(ras_poz,sl)
pohrani_sličice(poz_slič,geolok_slič)
neg_slič,geolok_slič=kreiraj_negativne_sličice_s_geolokacijama(ras_poz,sl)
pohrani_sličice(neg_slič,geolok_slič)
```

Kraj

Izračunavanje raspona sličica na kojima je prisutan atribut pripajanja (positivne sličice) radi se na sljedeći način:

- za svako pripajanje i njegovu geolokaciju pronađu se dvije najbliže geolokacije video snimke
- ukoliko je bliža geolokacija video snimke unutar praga udaljenosti pripajanju, uzima se da je pripajanje prisutno na tom dijelu video snimke
- na temelju udaljenosti geolokacija video snimke od geolokacije pripajanja i brzina kretanja automobila izračuna se točan raspon vremena video snimke na kojem se nalazi segment atributa pripajanja
- iz dobivenog raspona vremena izračuna se raspon sličica video snimke na kojem se nalazi segment atributa pripajanja

4. Eksperimenti i rezultati

Za potrebe ovog rada provedeno je nekoliko eksperimenata. Postupak učenja i testiranja u svakom eksperimentu je jednak, razlikuju se samo u vrsti ulaznih podataka (statične slike ili sekvene slike) te u korištenoj arhitekturi. U eksperimentima u kojima su korištene sekvene slike, duljina sekvene je eksperimentalno utvrđena i postavljena na 25 sličica.

4.1. Način učenja i testiranja u eksperimentima

Početni korak prilikom učenja jest normalizacija skupa podataka. Kako su podaci slike u boji, na podacima za treniranje se izračunava srednja vrijednost intenziteta piksela za svaki od kanala – crveni, zeleni i plavi. Dobivene srednje vrijednosti se zatim po kanalima oduzimaju svim podacima – podacima za treniranje, podacima za validaciju i podacima za testiranje.

Kako su sve arhitekture korištene za potrebe ovog rada zasnovane na naučenom prednjem dijelu arhitekture VGG-16, učenje se obavlja kroz 50 epoha i to na sljedeći način:

- 10 epoha učenja s početnom stopom učenja iznosa $5 \cdot 10^{-4}$ uz optimiranje samo novo nadodanih parametara – svi parametri naučenog prednjeg dijela arhitekture VGG-16 ne mijenjaju se tijekom ove faze
- 40 epoha učenja s početnom stopom učenja iznosa $1 \cdot 10^{-5}$ uz optimiranje kompletног skupa parametara (uključujući i parametre naučenog prednjeg dijela arhitekture VGG-16)
- korak učenja koristi gradijente izračunate nad mini-grupama veličina 5 ili 10, ovisno o memorijskoj zahtjevnosti modela
- optimizator korišten za učenje jest Adam optimizator

- stopa učenja se svakim korakom učenja eksponencijalno smanjuje:

$$\eta_t = \eta_0 \cdot 0.96^{\frac{t}{k}}, \quad (4.1)$$

gdje je t indeks koraka učenja, a k je broj koraka učenja u jednoj epohi

- svi novo nadodani slobodni parametri (isključujući parametre prednaučenog prednjeg dijela arhitekture VGG-16) inicijalizirani su na sljedeći način:
 - parametri slojeva (isključujući parametre pomaka) s po dijelovima linearnom aktivacijskom funkcijom inicijalizirani su koristeći inicijalizator koji skalira varijancu prema [10]
 - parametri slojeva (isključujući parametre pomaka) s normalizirajućom eksponencijalnom aktivacijskom funkcijom inicijalizirani su koristeći Xavier inicijalizaciju prema [10]
 - parametri pomaka svih slojeva inicijalizirani su s vrijednostima 0
- slobodni parametri regularizirani su L2 normom parametara s faktorom regularizacije iznosa $5 \cdot 10^{-4}$
- podskup za validaciju korišten je za odabir slobodnih parametara modela
 - na kraju svake epohe mjere se performanse modela na podskupu za validaciju
 - za testiranje se odabire onaj skup parametara koji postiže najbolju performansu na podskupu za validaciju – ovime se efektivno obavlja regularizacija (rano zaustavljanje) jer odabiremo onaj skup parametara koji najbolje generalizira

Svi eksperimenti provedeni su na nVIDIA grafičkim karticama GTX 1070 i GTX Titan. Programska izvedba ostvarena je koristeći programski jezik Python i programski okvir TensorFlow [5]. TensorFlow je vrlo popularan programski okvir za zadatke dubokog učenja jer pruža mogućnost automatske diferencijacije što omogućuje vrlo brzu izgradnju modela i provođenje eksperimenata.

Zadatak ovog rada definiran je kao binarna klasifikacija pa možemo iskoristiti 4 definirana tipa predikcija modela:

- TP (engl. *true positive*) – slika koja je označena kao pozitivna (atribut pripajanja je prisutan) i za koju je predikcija modela također takva
- TN (engl. *true negative*) – slika koja je označena kao negativna (atribut pripajanja nije prisutan) i za koju je predikcija modela također takva

- FP (engl. *false positive*) – slika koja je označena kao negativna, a predikcija modela je pozitivna
- FN (engl. *false negative*) – slika koja je označena kao pozitivna, a predikcija modela je negativna

Mjere koje su uzimane u obzir prilikom testiranja jesu točnost (engl. *accuracy*), preciznost (engl. *precision*) i odziv (engl. *recall*). Točnost jest definirana na sljedeći način:

$$\text{točnost} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (4.2)$$

Odziv jest definirana na sljedeći način:

$$\text{odziv} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4.3)$$

Preciznost jest definirana na sljedeći način:

$$\text{preciznost} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4.4)$$

Kako je izlaz modela vjerojatnost da je atribut pripajanja prisutan na slici, odnosno da atribut pripajanja nije prisutan na slici, definira se prag kao iznos vjerojatnosti iznad koje se smatra da je atribut prisutan. Za različite pravove dobivaju se različite vrijednosti preciznosti i odziva modela prilikom evaluacije. Ucrtavanjem svih točaka koje povezuju vrijednost preciznosti i odziva za različite pravove dobivamo graf ovisnosti preciznosti i odziva. Prosječna preciznost jest definirana kao površina ispod tog grafa.

4.2. Klasifikacija ručno označenih slika na različitim rezolucijama

Eksperimenti na ručno označenim slikama provedeni su koristeći arhitekturu za klasifikaciju pojedinačnih slika. Kako su eksperimenti nad sekvencama slika memorijski mnogo zahtjevniji od eksperimenata na pojedinačnim slikama, svrha ovog eksperimenta bila je utvrditi na kojoj rezoluciji ulaznih slika model gubi ekspresivnu moć i ne uspijeva dati zadovoljavajuću performansu.

Za početnu, originalnu rezoluciju pojedinačnih slika odabrana je rezolucija 700x280. Eksperiment je proveden redom na sljedećim rezolucijama: 700x280, 525x210, 350x140 i 175x70. Rezultati su dani u nastavku.

4.2.1. Rezultati na rezoluciji 700x280

Konačni rezultati dobiveni za eksperiment s ulaznim podacima rezolucije 700x280 su sljedeći:

Tablica 4.1: Statistički rezultati na rezoluciji 700x280

podskup podataka	točnost	preciznost	odziv	prosječna preciznost
podskup za učenje	1.0	1.0	1.0	1.0
podskup za validaciju	0.91	0.98	0.83	0.99
podskup za testiranje	0.95	0.99	0.91	0.99

Tablica 4.2: Odnos predikcija modela i stvarnih oznaka na rezoluciji 700x280 na podskupu za učenje

		predikcija modela	
		0	1
stvarna oznaka	0	898	0
	1	0	902

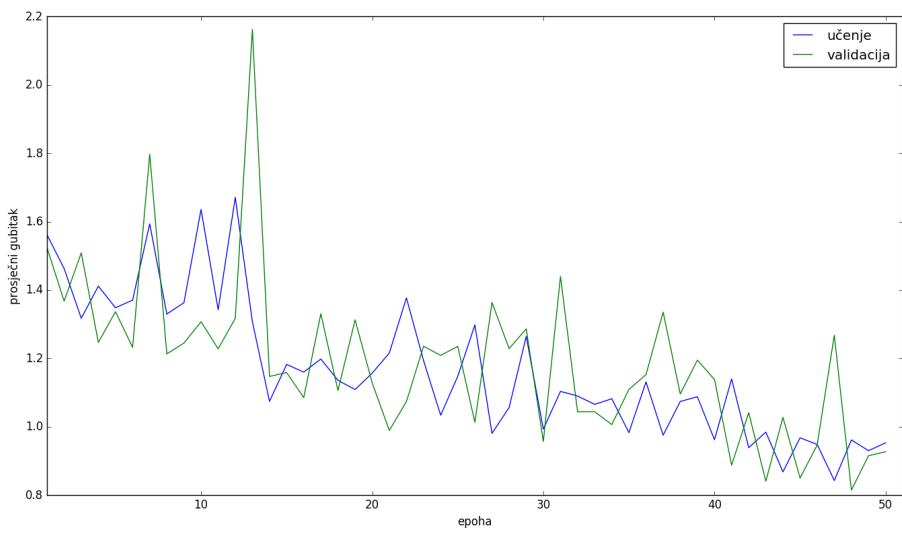
Tablica 4.3: Odnos predikcija modela i stvarnih oznaka na rezoluciji 700x280 na podskupu za validaciju

		predikcija modela	
		0	1
stvarna oznaka	0	308	5
	1	54	259

Tablica 4.4: Odnos predikcija modela i stvarnih oznaka na rezoluciji 700x280 na podskupu za testiranje

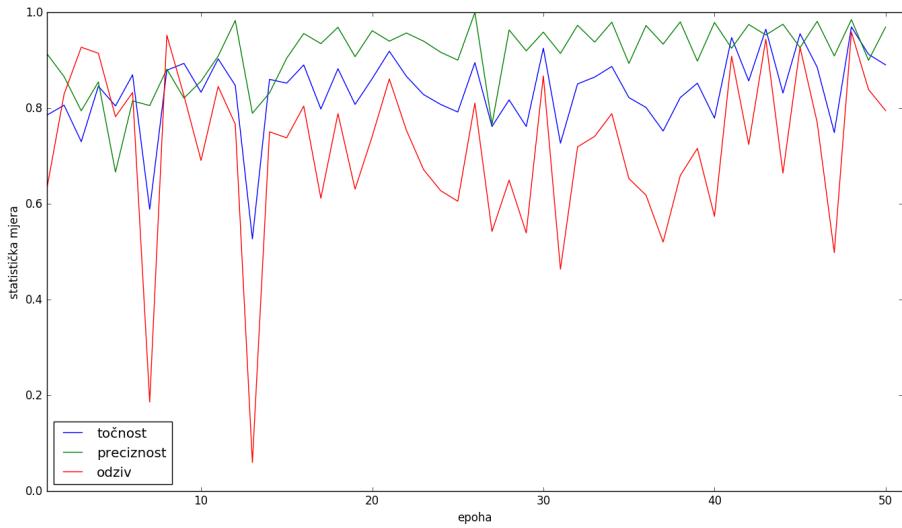
		predikcija modela	
		0	1
stvarna oznaka	0	295	2
	1	26	271

Tablica 4.1 prikazuje rezultate dobivene evaluacijom modela koji je postigao najbolju performansu na podskupu za validaciju prilikom učenja. Tablice 4.2, 4.3 i 4.4 prikazuju odnos predikcija modela i stvarnih oznaka kroz matrice zabune.



Slika 4.1: Kretanje prosječnog gubitka na podskupu za učenje i validaciju po epohama

Na slici 4.1 vidljivo je kretanje prosječnog gubitka po epohama na podskupu za učenje i na podskupu za validaciju.



Slika 4.2: Kretanje točnosti, preciznosti i odziva na podskupu za validaciju po epohama

Na slici 4.2 vidljivo je kretanje točnosti, preciznosti i odziva po epohama na podskupu za validaciju.

4.2.2. Rezultati na rezoluciji 525x210

Konačni rezultati dobiveni za eksperiment s ulaznim podacima rezolucije 525x210 su sljedeći:

Tablica 4.5: Statistički rezultati na rezoluciji 525x210

podskup podataka	točnost	preciznost	odziv	prosječna preciznost
podskup za učenje	1.0	1.0	1.0	1.0
podskup za validaciju	0.92	0.99	0.85	0.99
podskup za testiranje	0.98	0.99	0.96	0.99

Tablica 4.6: Odnos predikcija modela i stvarnih oznaka na rezoluciji 525x210 na podskupu za učenje

		predikcija modela	
		0	1
stvarna oznaka	0	898	0
	1	0	898

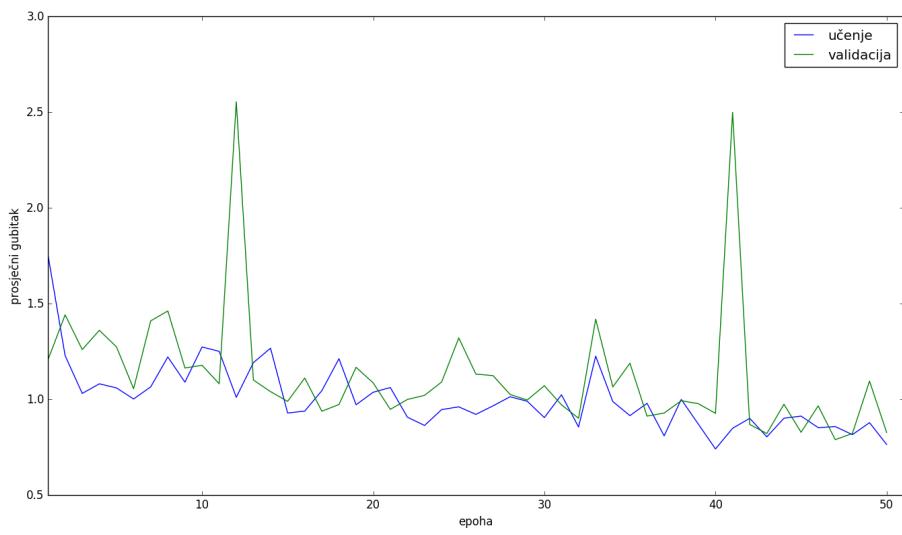
Tablica 4.7: Odnos predikcija modela i stvarnih oznaka na rezoluciji 525x210 na podskupu za validaciju

		predikcija modela	
		0	1
stvarna oznaka	0	310	3
	1	48	265

Tablica 4.8: Odnos predikcija modela i stvarnih oznaka na rezoluciji 525x210 na podskupu za testiranje

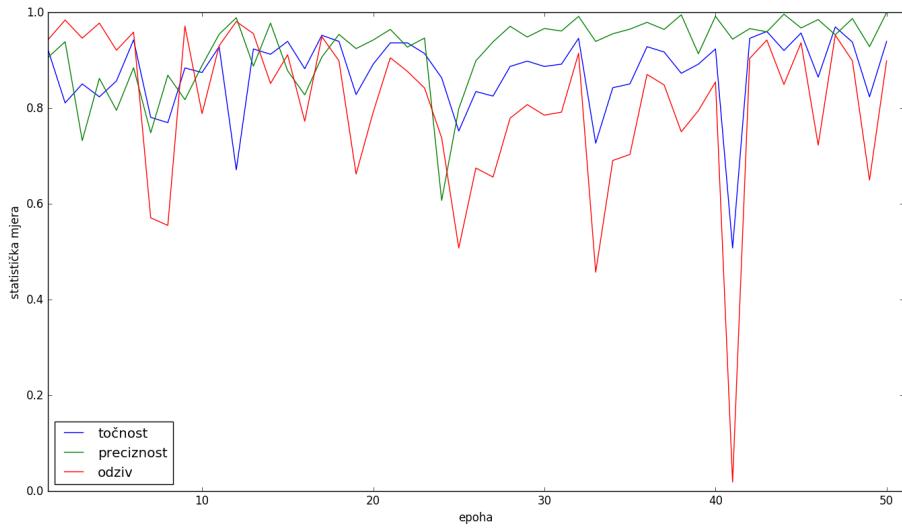
		predikcija modela	
		0	1
stvarna oznaka	0	295	2
	1	12	285

Tablica 4.5 prikazuje rezultate dobivene evaluacijom modela koji je postigao najbolju performansu na podskupu za validaciju prilikom učenja. Tablice 4.6, 4.7 i 4.8 prikazuju odnos predikcija modela i stvarnih oznaka kroz matrice zabune.



Slika 4.3: Kretanje prosječnog gubitka na podskupu za učenje i validaciju po epohama

Na slici 4.3 vidljivo je kretanje prosječnog gubitka po epohama na podskupu za učenje i na podskupu za validaciju.



Slika 4.4: Kretanje točnosti, preciznosti i odziva po epohama

Na slici 4.4 vidljivo je kretanje točnosti, preciznosti i odziva po epohama na podskupu za validaciju.

4.2.3. Rezultati na rezoluciji 350x140

Konačni rezultati dobiveni za eksperiment s ulaznim podacima rezolucije 350x140 su sljedeći:

Tablica 4.9: Statistički rezultati na rezoluciji 350x140

podskup podataka	točnost	preciznost	odziv	prosječna preciznost
podskup za učenje	1.0	1.0	1.0	1.0
podskup za validaciju	0.92	0.96	0.88	0.99
podskup za testiranje	0.98	0.96	0.99	0.99

Tablica 4.10: Odnos predikcija modela i stvarnih oznaka na rezoluciji 350x140 na podskupu za treniranje

		predikcija modela	
		0	1
stvarna oznaka	0	898	0
	1	0	898

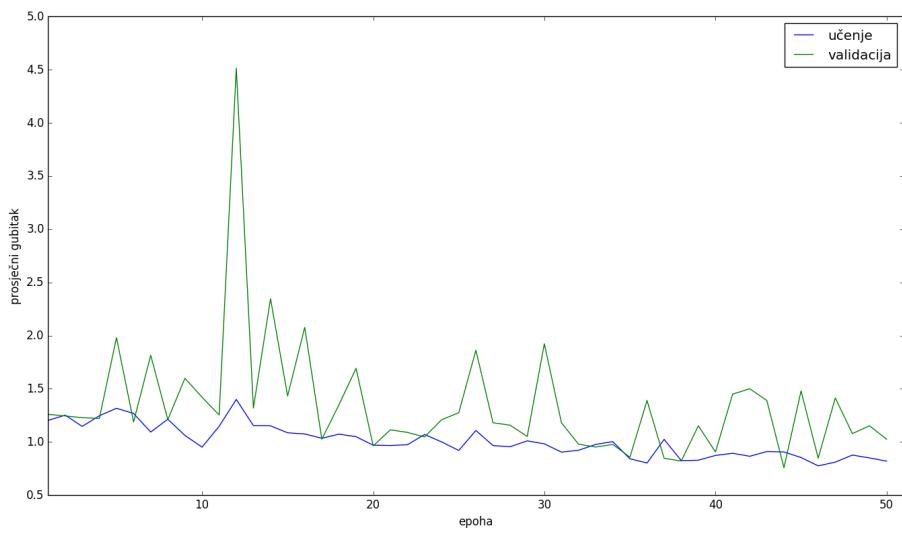
Tablica 4.11: Odnos predikcija modela i stvarnih oznaka na rezoluciji 350x140 na podskupu za validaciju

		predikcija modela	
		0	1
stvarna oznaka	0	303	10
	1	39	274

Tablica 4.12: Odnos predikcija modela i stvarnih oznaka na rezoluciji 350x140 na podskupu za testiranje

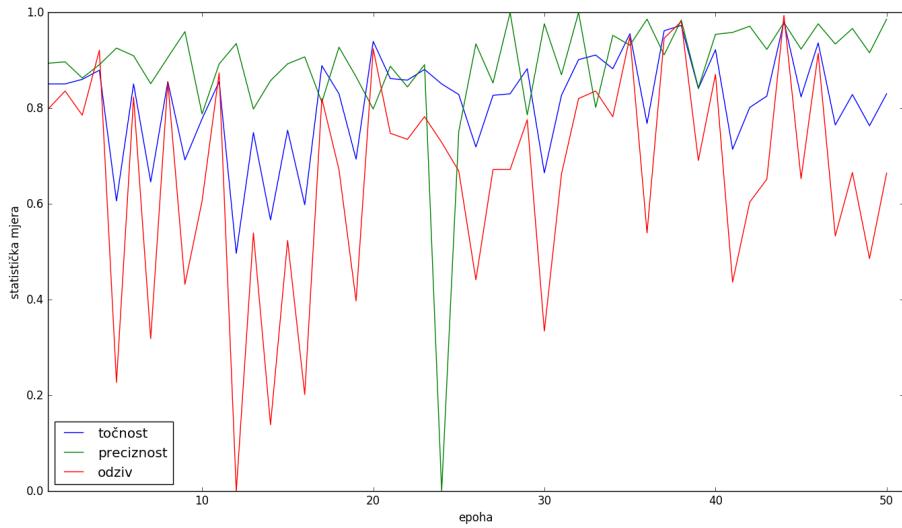
		predikcija modela	
		0	1
stvarna oznaka	0	286	11
	1	2	295

Tablica 4.9 prikazuje rezultate dobivene evaluacijom modela koji je postigao najbolju performansu na podskupu za validaciju prilikom učenja. Tablice 4.10, 4.11 i 4.12 prikazuju odnos predikcija modela i stvarnih oznaka kroz matrice zabune.



Slika 4.5: Kretanje prosječnog gubitka na podskupu za učenje i validaciju po epohama

Na slici 4.5 vidljivo je kretanje prosječnog gubitka po epohama na podskupu za učenje i na podskupu za validaciju.



Slika 4.6: Kretanje točnosti, preciznosti i odziva na podskupu za validaciju po epohama

Na slici 4.6 vidljivo je kretanje točnosti, preciznosti i odziva po epohama na podskupu za validaciju.

4.2.4. Rezultati na rezoluciji 175x70

Konačni rezultati dobiveni za eksperiment s ulaznim podacima rezolucije 175x70 su sljedeći:

Tablica 4.13: Statistički rezultati na rezoluciji 175x70

podskup podataka	točnost	preciznost	odziv	prosječna preciznost
podskup za učenje	1.0	1.0	1.0	1.0
podskup za validaciju	0.81	0.98	0.63	0.96
podskup za testiranje	0.71	0.96	0.43	0.92

Tablica 4.14: Odnos predikcija modela i stvarnih oznaka na rezoluciji 175x70 na podskupu za učenje

		predikcija modela	
		0	1
stvarna oznaka	0	898	0
	1	0	898

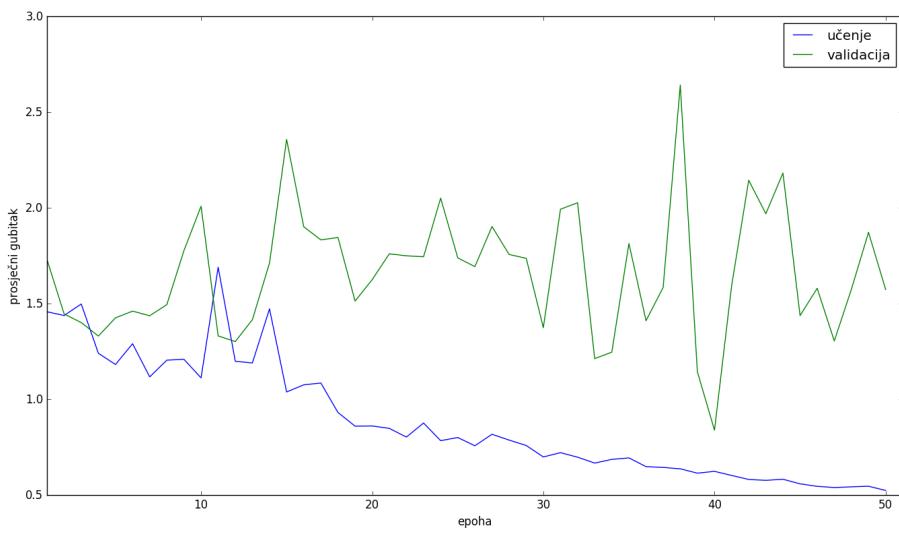
Tablica 4.15: Odnos predikcija modela i stvarnih oznaka na rezoluciji 175x70 na podskupu za validaciju

		predikcija modela	
		0	1
stvarna oznaka	0	310	3
	1	117	196

Tablica 4.16: Odnos predikcija modela i stvarnih oznaka na rezoluciji 175x70 na podskupu za testiranje

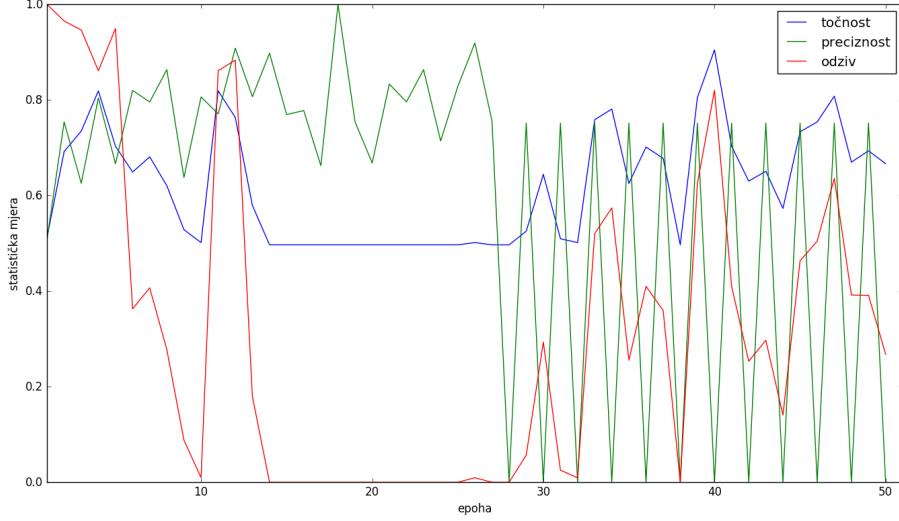
		predikcija modela	
		0	1
stvarna oznaka	0	291	6
	1	169	128

Tablica 4.13 prikazuje rezultate dobivene evaluacijom modela koji je postigao najbolju performansu na podskupu za validaciju prilikom učenja. Tablice 4.14, 4.15 i 4.16 prikazuju odnos predikcija modela i stvarnih oznaka kroz matrice zabune.



Slika 4.7: Kretanje prosječnog gubitka na skupu za učenje i validaciju po epohama

Na slici 4.7 vidljivo je kretanje prosječnog gubitka po epohama na podskupu za učenje i na podskupu za validaciju.



Slika 4.8: Kretanje točnosti, preciznosti i odziva po epohama

Na slici 4.8 vidljivo je kretanje točnosti, preciznosti i odziva po epohama na podskupu za validaciju.

Vidljivo je kako se tek na rezoluciji 175x70 pojavljuje značajniji pad performansi. Iz tog razloga se za eksperimente nad sekvencama slika (zbog veće memorijske zah-tjevnosti) koriste rezolucije pojedinih slika 350x140.

4.3. Klasifikacija pojedinačnih slika označenih konzistentno s projektom FTTS iRAP na originalnoj rezoluciji

Za razliku od prethodnih eksperimenata, ovaj eksperiment je proveden na skupu podataka označenom konzistentno sa sustavom FTTS iRAP. Eksperiment je proveden na pojedinačnim slikama rezolucije 700x280 kako bi se u dalnjim eksperimentima moglo usporediti poboljšava li se rezultat uvođenjem dodatnih informacija sa sekvencom slika koje prethode slici za koju je određena oznaka prisutnosti atributa pripajanja.

Tablica 4.17: Statistički rezultati

podskup podataka	točnost	preciznost	odziv	prosječna preciznost
podskup za učenje	0.95	0.95	0.95	0.99
podskup za validaciju	0.88	0.92	0.83	0.93
podskup za testiranje	0.82	0.87	0.76	0.90

Tablica 4.18: Odnos predikcija modela i stvarnih oznaka na podskupu za učenje

stvarna oznaka \ predikcija modela	0	1
0	3572	205
1	173	3604

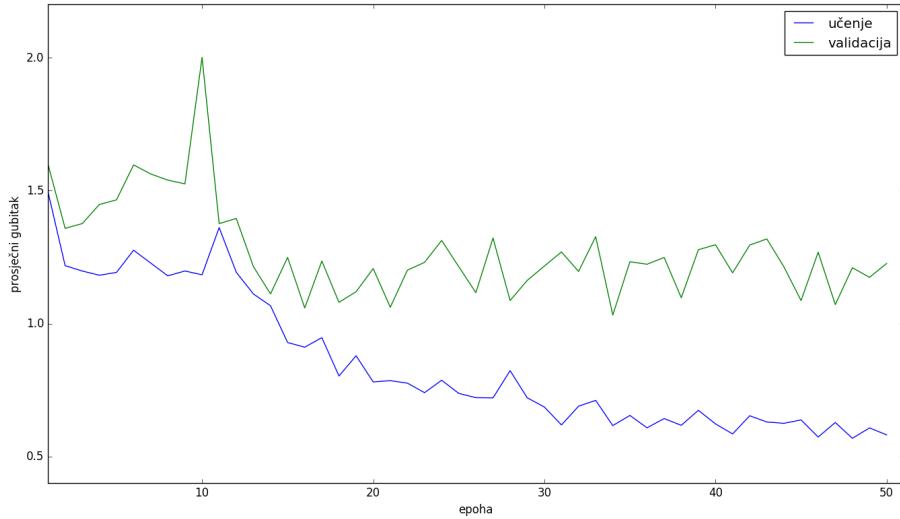
Tablica 4.19: Odnos predikcija modela i stvarnih oznaka na podskupu za validaciju

stvarna oznaka \ predikcija modela	0	1
0	800	60
1	149	711

Tablica 4.20: Odnos predikcija modela i stvarnih oznaka na podskupu za testiranje

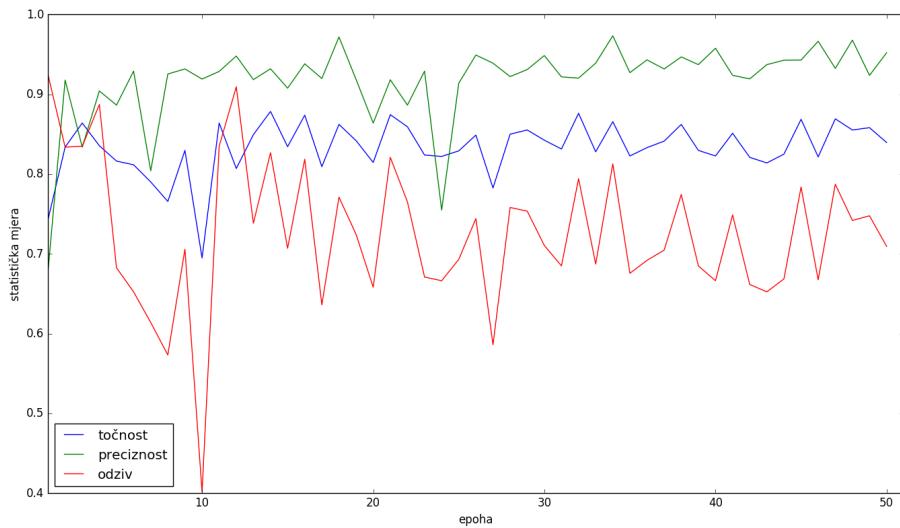
stvarna oznaka \ predikcija modela	0	1
0	731	90
1	198	623

Tablica 4.17 prikazuje rezultate dobivene evaluacijom modela koji je postigao najbolju performansu na podskupu za validaciju prilikom učenja. Tablice 4.18, 4.19 i 4.20 prikazuju odnos predikcija modela i stvarnih oznaka kroz matrice zabune.



Slika 4.9: Kretanje prosječnog gubitka na skupu za učenje i validaciju po epohama

Na slici 4.9 vidljivo je kretanje prosječnog gubitka po epohama na podskupu za učenje i na podskupu za validaciju.



Slika 4.10: Kretanje točnosti, preciznosti i odziva po epohama

Na slici 4.10 vidljivo je kretanje točnosti, preciznosti i odziva po epohama na podskupu

za validaciju.

4.4. Klasifikacija sekvenci slika označenih konzistentno s projektom FTTS iRAP koristeći vremensko-prostorno sažimanje

Konačni rezultati dobiveni za eksperiment sa sekvencom slika rezolucije 350x140 su sljedeći:

Tablica 4.21: Statistički rezultati

podskup podataka	točnost	preciznost	odziv	prosječna preciznost
podskup za učenje	0.91	0.91	0.91	0.97
podskup za validaciju	0.87	0.90	0.85	0.95
podskup za testiranje	0.83	0.88	0.77	0.91

Tablica 4.22: Odnos predikcija modela i stvarnih oznaka na podskupu za učenje

predikcija modela		0	1
stvarna oznaka	0		
0		3430	347
1		338	3439

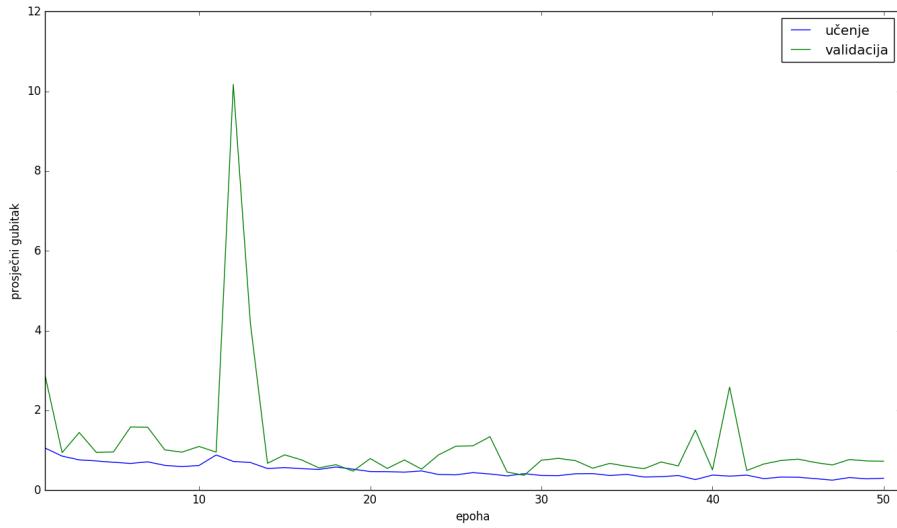
Tablica 4.23: Odnos predikcija modela i stvarnih oznaka na podskupu za validaciju

predikcija modela		0	1
stvarna oznaka	0		
0		775	85
1		133	727

Tablica 4.24: Odnos predikcija modela i stvarnih oznaka na podskupu za testiranje

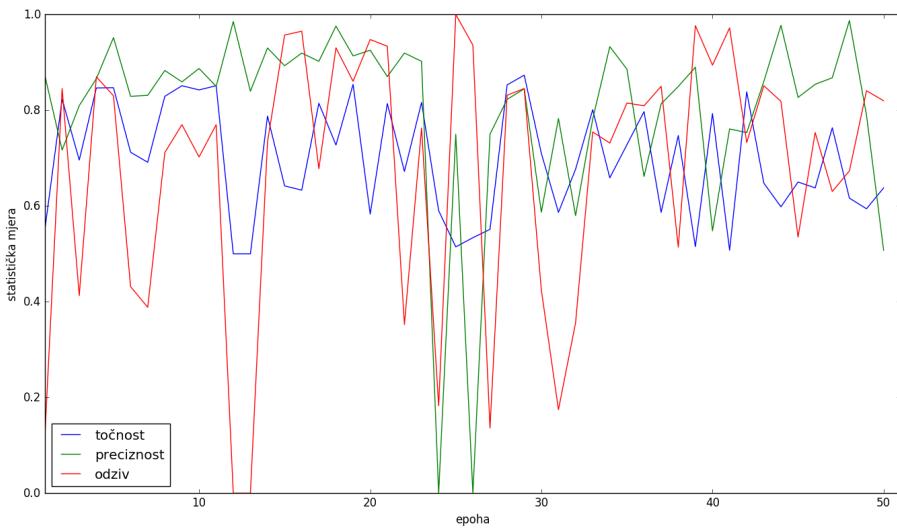
predikcija modela		0	1
stvarna oznaka	0		
0		734	87
1		189	632

Tablica 4.21 prikazuje rezultate dobivene evaluacijom modela koji je postigao najbolju performansu na podskupu za validaciju prilikom učenja. Tablice 4.22, 4.23 i 4.24 prikazuju odnos predikcija modela i stvarnih oznaka kroz matrice zabune.



Slika 4.11: Kretanje prosječnog gubitka na skupu za učenje i validaciju po epohama

Na slici 4.11 vidljivo je kretanje prosječnog gubitka po epohama na podskupu za učenje i na podskupu za validaciju.



Slika 4.12: Kretanje točnosti, preciznosti i odziva po epohama

Na slici 4.12 vidljivo je kretanje točnosti, preciznosti i odziva po epohama na podskupu

za validaciju.

4.5. Klasifikacija sekvenci slika označenih konzistentno s projektom FTTS iRAP koristeći povratne LSTM celije

Konačni rezultati dobiveni za eksperiment sa sekvenčnim ulaznim podacima rezolucije 350x140 su sljedeći:

Tablica 4.25: Statistički rezultati

podskup podataka	točnost	preciznost	odziv	prosječna preciznost
podskup za učenje	0.99	0.99	0.99	0.99
podskup za validaciju	0.89	0.95	0.83	0.94
podskup za testiranje	0.87	0.91	0.82	0.94

Tablica 4.26: Odnos predikcija modela i stvarnih oznaka na podskupu za učenje

predikcija modela		0	1
stvarna oznaka	0	3731	46
1	3	3774	

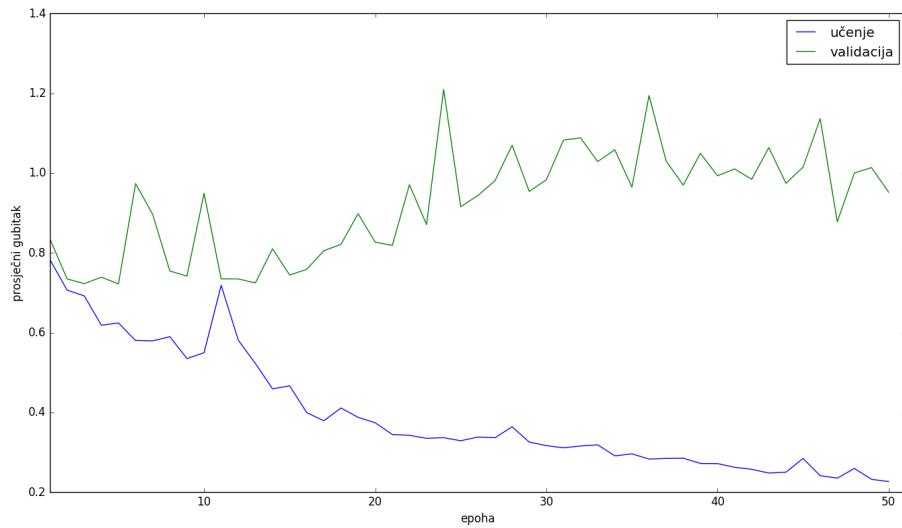
Tablica 4.27: Odnos predikcija modela i stvarnih oznaka na podskupu za validaciju

predikcija modela		0	1
stvarna oznaka	0	821	39
1	147	713	

Tablica 4.28: Odnos predikcija modela i stvarnih oznaka na podskupu za testiranje

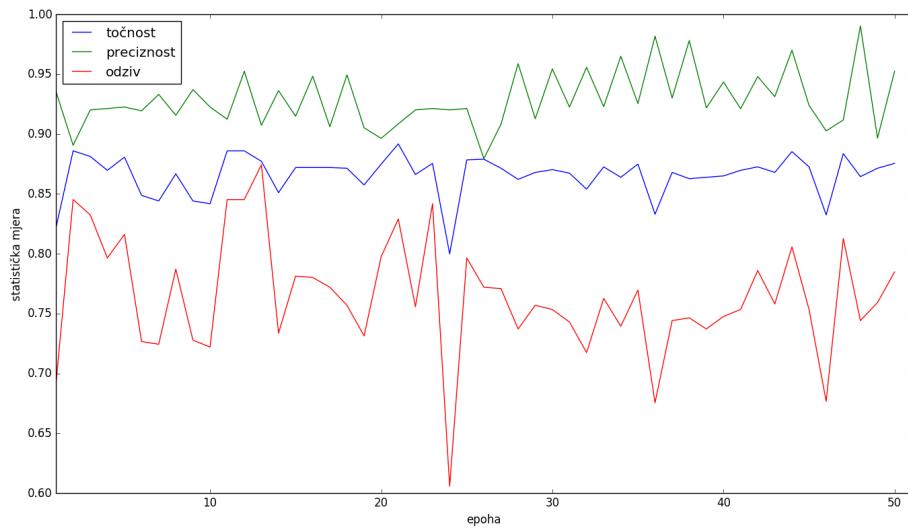
predikcija modela		0	1
stvarna oznaka	0	735	68
1	149	672	

Tablica 4.25 prikazuje rezultate dobivene evaluacijom modela koji je postigao najbolju performansu na podskupu za validaciju prilikom učenja. Tablice 4.26, 4.27 i 4.28 prikazuju odnos predikcija modela i stvarnih oznaka kroz matrice zabune.



Slika 4.13: Kretanje prosječnog gubitka na skupu za učenje i validaciju po epohama

Na slici 4.13 vidljivo je kretanje prosječnog gubitka po epohama na podskupu za učenje i na podskupu za validaciju.



Slika 4.14: Kretanje točnosti, preciznosti i odziva po epohama

Na slici 4.14 vidljivo je kretanje točnosti, preciznosti i odziva po epohama na podskupu

za validaciju.

4.6. Klasifikacija sekvenci slika označenih konzistentno s projektom FTTS iRAP koristeći vremenski potpuno povezani sloj

Konačni rezultati dobiveni za eksperiment sa sekvencijalnim ulaznim podacima rezolucije 350x140 su sljedeći:

Tablica 4.29: Statistički rezultati

podskup podataka	točnost	preciznost	odziv	prosječna preciznost
podskup za učenje	0.99	0.99	1.0	1.0
podskup za validaciju	0.89	0.98	0.80	0.96
podskup za testiranje	0.86	0.95	0.76	0.94

Tablica 4.30: Odnos predikcija modela i stvarnih oznaka na podskupu za učenje

stvarna oznaka \ predikcija modela	0	1
0	3775	2
1	0	3777

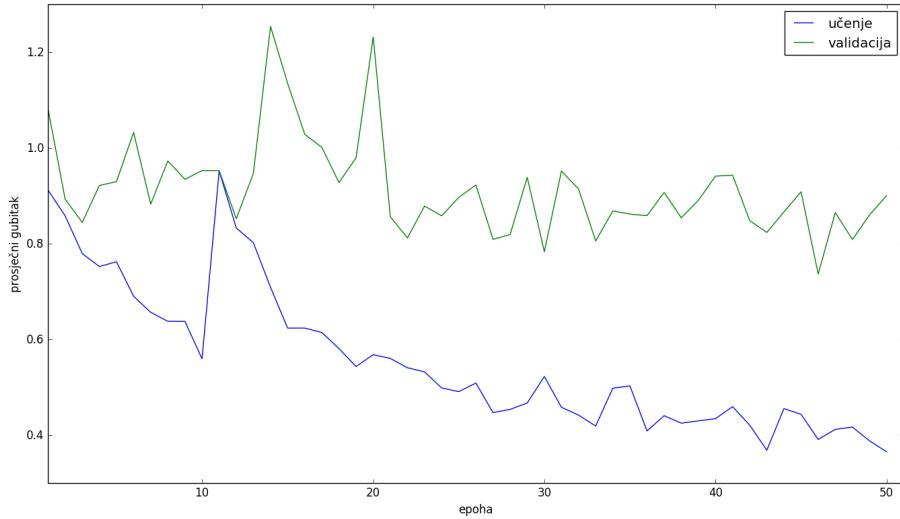
Tablica 4.31: Odnos predikcija modela i stvarnih oznaka na podskupu za validaciju

stvarna oznaka \ predikcija modela	0	1
0	843	17
1	176	684

Tablica 4.32: Odnos predikcija modela i stvarnih oznaka na podskupu za testiranje

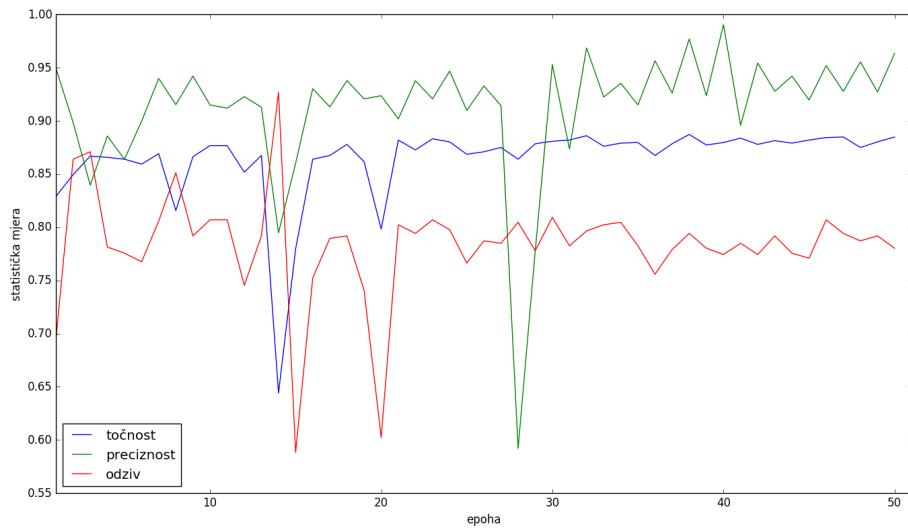
stvarna oznaka \ predikcija modela	0	1
0	787	34
1	199	622

Tablica 4.29 prikazuje rezultate dobivene evaluacijom modela koji je postigao najbolju performansu na podskupu za validaciju prilikom učenja. Tablice 4.30, 4.31 i 4.32 prikazuju odnos predikcija modela i stvarnih oznaka kroz matrice zabune.



Slika 4.15: Kretanje prosječnog gubitka na skupu za učenje i validaciju po epohama

Na slici 4.15 vidljivo je kretanje prosječnog gubitka po epohama na podskupu za učenje i na podskupu za validaciju.



Slika 4.16: Kretanje točnosti, preciznosti i odziva po epohama

Na slici 4.16 vidljivo je kretanje točnosti, preciznosti i odziva po epohama na podskupu

za validaciju.

4.7. Analiza rezultata

Prvi dio eksperimenata na ručno označenim slikama proveden je kako bi se ustanovila rezolucija slika na kojima je još uvijek moguće prepoznati diskriminativni dio. Bilo je važno znati granicu do koje je moguće smanjiti slike jer su eksperimenti nad sekvencama slika memorijski mnogo zahtjevniji pa je za njihovo učenje s kraja na kraj potrebno imati slike mnogo manjih rezolucija.

Drugi dio eksperimenata proveden je na slikama označenima konzistentno sa sustavom FTTS iRAP. Proveden je eksperiment na pojedinačnim i na sekvencama slika kako bi se ustanovilo dobiva li se na točnosti uvođenjem dodatnih informacija kroz sekvencu slika koja prethodi označenoj slici.

4.7.1. Analiza rezultata na ručno označenim slikama

Eksperimentima na ručno označenim slikama utvrđeno je da je rezolucija slika 350x140 pogodna za korištenje u sekvencialnom pristupu jer je to najniža rezolucija na kojoj se ne događa pad performanse.

Pad performanse može se objasniti time što je prednji dio arhitekture VGG-16 koji služi za izlučivanje značajki originalno naučen na slikama rezolucije 224x224. Najveći pad performanse vidljiv je na mjeri odziva – zbog manje rezolucije slika model ne uspijeva uvijek prepoznati diskriminativni dio na temelju kojeg bi donio odluku da je slika pozitivna.

4.7.2. Analiza rezultata na slikama označenima konzistentno sa sustavom FTTS iRAP

Prvi dio eksperimenata na slikama označenima konzistentno sa sustavom FTTS iRAP proveden je nad pojedinačnim slikama. Vidljiv je značajniji pad performansi u odnosu na ručno označene slike. Razlog tomu jest način na koji su slike u sustavu FTTS iRAP označene, kako je objašnjeno u poglavljju 2.

Kako su eksperimenti pokazali, uvođenjem sekvence slika koja prethodi označenoj slici dobiva se na performansama u odnosu na pojedinačne slike označene konzistentno sa sustavom FTTS iRAP. Najbolji rezultat dobiven je modelom koji dobivene prostorne značajke obrađuje povratnim LSTM čelijama.

Performanse modela koji prostorne značajke slika označenih konzistentno sa sustavom FTTS iRAP obrađuje povratnim LSTM celijama i dalje su niža od performansa modela na ručno označenim pojedinačnim slikama. Pad performansa možemo analizirati tako da promotrimo sekvence slika iz podskupa za testiranje koje model loše klasificira. U nastavku su pokazani tipovi pogrešaka na kojima model grijesi.

Lažni pozitivi (engl. *false positives*)



Slika 4.17: Lažni pozitiv s lijevom trakom različite boje, preuzeto s [1]



Slika 4.18: Lažni pozitiv s diskriminativnim uzorcima kakvi se pojavljuju kod pripajanja, preuzeto s [1]

Slika 4.17 prikazuje cestu gdje je lijeva traka različite boje od ostalih. Model mnogo takvih sličnih primjera označava kao pozitivne. Moguć razlog je što ga različitost lijeve trake može podsjećati na pripajanje. Slika 4.18 prikazuje razdvajanje koje ima vrlo slične diskriminativne uzorke onima na pripajanjima. Modelu je iz tog razloga teško donijeti odluku je li riječ o pozitivu ili negativu.

Lažni negativi (engl. *false negatives*)



Slika 4.19: Lažni negativ s razdvajanjem službeno označenim kao pripajanje, preuzeto s [1]



Slika 4.20: Lažni negativ s pripajanjem na suprotnoj strani autoceste službeno označenim kao pripajanje, preuzeto s [1]



Slika 4.21: Lažni negativ s pripajanjem u sjeni, preuzeto s [1]

Slika 4.19 prikazuje situaciju u kojoj je razdvajanje, prema službenim oznakama sustava FTTS iRAP, označeno kao pripajanje. Slika 4.20 prikazuje situaciju u kojoj se dogodilo da je zbog prisutnosti pripajanja na drugoj strani autoceste i promatrana sekvenca označena kao pozitivna. Slika 4.21 prikazuje situaciju u kojoj je pripajanje u sjeni i model ne uspijeva prepoznati diskriminativnu značajku na temelju koje bi mogao prepoznati prisutnost atribut apripajanja.

U razgovoru s kreatorima sustava FTTS iRAP zaključeno je kako je moguće da postoje moguće nekonzistentnosti i kriva označavanja prisutnosti atributa pripajanja uslijed ljudske pogreške i uslijed načina na koji se oznake dodjeljuju. Iz tog razloga skup podataka sadrži "nečistoće" koje je potrebno validirati sa samim sustavom. Po-

datak koji se u tom slučaju može pokazati kao koristan je geolokacija svakog primjera iz skupa podataka za učenje. Na taj način moguće je relativno brzo provjeriti sve sumnjive primjere iz skupa za učenje, eventualno im promijeniti oznaku prisutnosti atributa te generirati novi, čišći skup podataka.

5. Zaključak

Organizacija iRAP bavi se inspekcijom kvaliteta ceste pridruživanjem različitih sigurnosnih atributa segmentima prometnica. Sustav FTTS iRAP je geolokacijski informacijski sustav koji sadrži video snimke s geolokacijskim oznakama i pridruženim sigurnosnim atributima prema naputcima organizacije iRAP. Označavanje segmenata video snimaka obavlja se ručno što može biti skupo, vremenski zahtjevno i otvara mogućnost za brojne pogreške.

Duboko učenje donijelo je revoluciju u mnogim područjima pa tako i u području računalnog vida. Cilj rada bio je iskoristiti moć dubokog učenja i pokušati napraviti sustav zasnovan na dubokom modelu koji bi iz video snimaka naučio automatski označavati jedan od velikog broja sigurnosnih atributa - pripajanje na autocesti. Prednost korištenja dubokog modela jest jeftinije i objektivnije pridruživanje sigurnosnih atributa segmentima video snimaka. Konačno naučen sustav postizao je zadovoljavajuću performanse s obzirom na nekonzistentnost oznaka skupa podataka. Naučen sustav može se iskoristiti za brzu validaciju ručno označenih atributa. Svaki uzorak koji model krivo označi potencijalno je pogrešno označen i u sustavu pa je iz njegove geolokacije moguće brzo validirati prisutnost atributa pripajanja u samom sustavu.

Prvi korak za budući rad i poboljšanje sustava bio bi validirati skup podataka sa sustavom FTTS iRAP i tako ukloniti svaku moguću nekonzistentnost koja dovodi do lošijih performansi. Nadalje, pokazano je kako korištenje sekvenci slika dovodi do povećanja performansi modela – poboljšanje bi se moglo ostvariti korištenjem mnogo duljih sekvenci uz rjeđe uzorkovanje te uz uzorkovanje slika koje prethode i koje slijede pojedinačnoj slici od interesa. Moguća poboljšanja mogla bi se ostvariti i u vidu korištenja modernijih naučenih arhitektura korištenih za tipične zadatke u računalnom vidu.

LITERATURA

- [1] <https://he.ftts-irap.org/>, . [pristupano 14. lipnja 2017.].
- [2] <https://www.google.hr/maps?source=tldsi&hl=en>, . [pristupano 16. lipnja 2017.].
- [3] iRAP Star Rating and Investment Plan Coding Manual. http://downloads.irap.org/docs/RAP-SR-2-2_Star_Rating_coding_manual.pdf, 2014. [pristupano 14. lipnja 2017.].
- [4] iRAP Methodology Fact Sheet #3 Road attributes. <http://www.irap.org/en/about-irap-3/methodology?download=132:irap-methodology-fact-sheet-3-road-attributes>, 2015. [pristupano 14. lipnja 2017.].
- [5] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yanqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, i Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <http://tensorflow.org/>. Software available from tensorflow.org.
- [6] Dan C. Ciresan, Ueli Meier, i Jürgen Schmidhuber. Multi-column deep neural networks for image classification. *CoRR*, abs/1202.2745, 2012. URL <http://arxiv.org/abs/1202.2745>.
- [7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, i L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. U *CVPR09*, 2009.

- [8] H. El Khiyari i H. Wechsler. Face Recognition across Time Lapse Using Convolutional Neural Networks. *Journal of Information Security*, 7(3):141–151, 2016.
- [9] Ian Goodfellow, Yoshua Bengio, i Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, i Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015. URL <http://arxiv.org/abs/1502.01852>.
- [11] Sergey Ioffe i Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.
- [12] Diederik P. Kingma i Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL <http://arxiv.org/abs/1412.6980>.
- [13] Joe Yue-Hei Ng, Matthew J. Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, i George Toderici. Beyond short snippets: Deep networks for video classification. *CoRR*, abs/1503.08909, 2015. URL <http://arxiv.org/abs/1503.08909>.
- [14] Maxime Oquab, Léon Bottou, Ivan Laptev, i Josef Sivic. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. U *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, United States, Lipanj 2014. URL <https://hal.inria.fr/hal-00911179>.
- [15] Ivan Relić. Klasifikacija slika dubokim konvolucijskim mrežama, 2016. seminarski rad.
- [16] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, i Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.
- [17] Dominik Scherer, Andreas Muller, i Sven Behnke. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. 2010.

- [18] Karen Simonyan i Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556, 2014. URL <http://arxiv.org/abs/1409.1556>.
- [19] Vedran Vukotić. Raspoznavanje objekata dubokim neuronskim mrežama, 2014. diplomski rad.
- [20] Wojciech Zaremba, Ilya Sutskever, i Oriol Vinyals. Recurrent neural network regularization. *CoRR*, abs/1409.2329, 2014. URL <http://arxiv.org/abs/1409.2329>.
- [21] Siniša Šegvić. Uvodno predavanje. <http://www.zemris.fer.hr/~ssegvic/du/du0intro.pdf>, 2016. [pristupano 17. lipnja 2017.].
- [22] Jan Šnajder i Bojana Dalbelo Bašić. *Strojno učenje*. 2014.

Detekcija sigurnosnih atributa prometnica u snimkama

Sažetak

Ovaj rad opisuje modeliranje i učenje dubokih modela za klasifikaciju sekvenci prometnih scena. Svrha učenje modela za klasifikaciju sekvenci prometnih scena jest automatizirati dodjeljivanje sigurnosnih atributa definiranih od strane organizacije iRAP. Dodjeljivanje sigurnosnih atributa u različitim sustavima trenutno se obavlja ručno. Korištenje dubokih modela pruža jeftiniju i objektivniju alternativu takvom zadatku.

Predloženo je nekoliko arhitektura koje rade s pojedinačnim slikama i sa sekvenčama slika te koje je moguće učiti s kraja na kraj. Skupovi podataka generirani su na dva načina – ručnim označavanjem i automatskom metodom koja oznake generira na temelju podataka iz sustava FTTS iRAP. Prikazani su eksperimentalni rezultati dobiveni nad oba skupa. Ručno označen skup podataka korišten je za pronalaženje najniže rezolucije na kojoj se ručno označene pojedinačne slike i dalje klasificiraju uspješno. Pronalazak niske rezolucije slika na kojoj klasifikacija i dalje uspješno radi važna je jer učenje nad sekvenčama slika zahtjeva više memorije nego učenje na pojedinačnim slikama. Skup podataka označen na temelju podataka iz sustava FTTS iRAP korišten je za usporedbu dobitka na performansama uvođenjem dodatnih informacija kroz sekvencu slika.

Ključne riječi: duboko učenje, sekvencijalni podaci, iRAP sigurnosni atributi, klasifikacija, prometne scene

Detection of road-assessment attributes in video

Abstract

This paper describes modeling and training of deep models for classification of traffic scene sequences. Purpose of training the model for classifications of traffic scenes is to automate the assignment of safety attributes defined by the iRAP organization. Assignment of safety attributes in different systems is being performed manually at the moment. Usage of the deep models provides cheaper and more objective alternative for that task.

Several architectures that work with single images and image sequences and which can be trained end-to-end are suggested. Datasets are generated in two ways – by manual labeling and by the automated method which generates labels based on the data from FTTS iRAP system. Experimental results on both datasets are presented. Manually labeled dataset is used to find the lowest resolution on which manually labeled single images are still successfully classified. Determining low image resolution on which classification still works successfully is important because training on image sequences has greater memory requirements than training on single images. Dataset with labels based on the data from FTTS iRAP system is used to compare performance gain with introducing the additional information through the image sequences.

Keywords: deep learning, sequence data, iRAP safety attributes, classification, traffic scenes