SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1725

Konvolucijske reprezentacije za dohvat slike na temelju sadržaja

Filip Jakov Bulić

Voditelj: Siniša Šegvić

Zagreb, lipanj 2018.

Sadržaj

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

Zagreb, 16. ožujka 2018.

DIPLOMSKI ZADATAK br. 1725

Pristupnik: **Filip Jakov Bulić (0036479748)**
Studij: Računarstvo
Profil: Računarska znanost

Zadatak: **Konvolucijske reprezentacije za dohvat slike na temelju sadržaja**

Opis zadatka:

Pretraživanje slikovnih baza prema slikovnom sadržaju je važan zadatak računalnog vida. Posebno su zanimljivi pristupi koji taj zadatak ostvaruju na temelju slikovnog primjera kojeg prilaže korisnik. U posljednje vrijeme, veliki uspjeh u tom području ostvaruju pristupi temeljeni na naučenim značajkama do kojih dolazimo dubokim konvolucijskim modelima.
U okviru rada, potrebno je proučiti i ukratko opisati postojeće pristupe dohvata slika prema zadanom slikovnom primjeru. Uhodati postupke nadziranog učenja prikladnih slikovnih reprezentacija. Validirati hiperparametre te prikazati i ocijeniti ostvarene rezultate na slikovnom skupu Oxford5k. Komentirati primjenjivost postupaka na kolekcije s velikim brojem primjera za učenje. 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: 16. ožujka 2018.
Rok za predaju rada: 29. lipnja 2018.

Mentor:
Prof. dr. sc. Siniša Šegvić

Djelovođa:
Doc. dr. sc. Tomislav Hrkać

Predsjednik odbora za diplomski rad profila:
Prof. dr. sc. Siniša Srbljić

Zahvaljujem mentoru prof. dr. sc. Siniši Šegviću na stručnoj pomoći i savjetima kroz obrazovanje tijekom studija

1. Uvod

Računalni vid je interdisciplinarno područje koje se bavi prikupljanjem, obradom, analiziranjem i razumijevanjem višedimenzionalnih podataka (slika) s ciljem izvlačenja korisnih numeričkih ili simboličkih podataka. Računalni vid pokriva područja poput detekcije, prepoznavanja i praćenja objekata, semantičke segmentacije i mnogih drugih.

Zbog eksplozije količine podataka koja je dostupna u digitalnom obliku i brzog razvoja računala u zadnjih nekoliko godina počele su se razvijati velike neuronske mreže s mnoštvom slojeva: duboke neuronske mreže (engl. d*eep neural networks*). Pozitivne strane takvog pristupa uključuju prilagođenost naučenih značajki konkretnom problemu i njegovom skupu uzoraka, dijeljenje značajki između više klasa... Posebno su interesantne duboke konvolucijske neuronske mreže koje su pronašle mnoge aplikacije u svijetu računalnog vida. Napreci u arhitekturama pružaju poboljšanja u brzini i točnosti riješavanja mnogih problema te su od temeljne važnosti za računalni vid u cjelini.

Pretraživanje slikovnih baza prema slikovnom sadržaju je važan zadatak računalnog vida. Posebno su zanimljivi pristupi koji taj zadatak ostvaruju na temelju slikovnog primjera kojeg prilaže korisnik. U posljednje vrijeme, veliki uspjeh u tom području ostvaruju pristupi temeljeni na naučenim značajkama do kojih dolazimo dubokim konvolucijskim modelima.

Ovaj rad kroz nekoliko dijelova opisuje arhitekturu korištene mreže, implementaciju te dobivene rezultate. Prvi dio rada opisuje osnovne dijelove dubokih konvolucijskih mreža te metode koje koristimo kako bi takve mreže izgradili. Drugi dio opisuje dijelove sustava za pretraživanje slikovnih baza prema slikovnom sadržaju. Nakon toga slijedi opis korištenih skupova podataka, implementacijski detalji modela, provedeni eksperimenti i dobiveni rezultati prilikom korištenja opisanog modela. Konačno, zaključak daje buduće smjerove rada i moguća poboljšanja.

2. Duboke konvolucijske neuronske mreže

U ovom poglavlju biti će objašnjeni temeljni elementi konvolucijskih neuronskih mreža. Za detaljnija objašnjenja preporučuje se proučiti stručnu literaturu.

2.1 Umjetne neuronske mreže i duboko učenje

Duboko učenje kao grana strojnog učenja zabilježila je brojne uspjehe u području računalnog vida. Razvojem područja počeli su se koristiti novi modeli neuronskih mreža te robusniji algoritmi pronalaženja optimalnih parametara za pojedine zadatke i arhitekture. Općenito, neuronske mreže su modeli koji se sastoje od skupa međusobno povezanih osnovnih jedinica – neurona. Izlaz jednog neurona funkcija je ulaza i parametara (težina) neurona.

$$y = f\left(\vec{x} \cdot \vec{w} + b\right) = f\left(\sum_{i=1}^{n} x_i \cdot w_i + b\right) \tag{2.1}$$

Ovime je definiran najjednostavniji oblik neurona, gdje težine predstavljaju koliko određeni ulaz doprinosi u odluci o aktiviranju neurona.

Najzastupljeniji način organizacije neuronskih mreža su slojevi, odnosno, lančano povezane grupe neurona strukturirane na način da izlaz promatranog sloja predstavlja ulaz u sljedeći sloj. Ako je prvi sloj definiran kao:

$$h^{(1)} = f^{(1)}\left(x \cdot W^{(1)} + b^{(1)}\right) \tag{2.2}$$

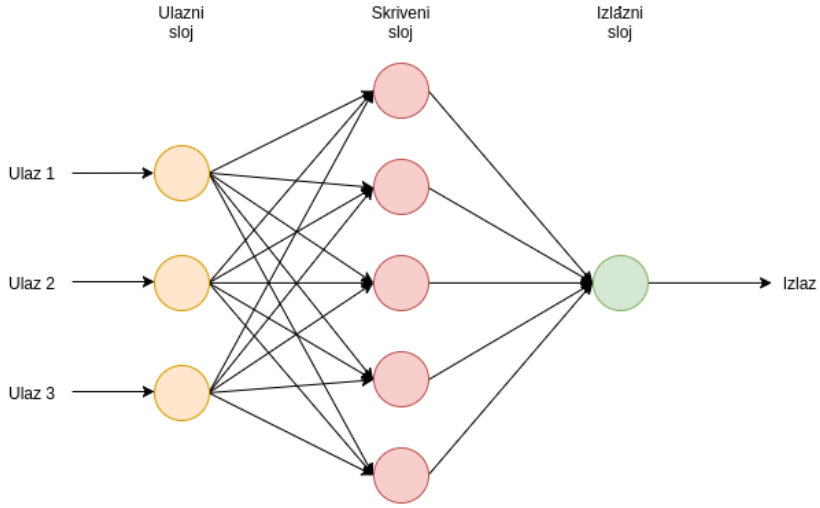
sljedeći sloj definiran je kao:

$$h^{(2)} = f^{(2)}\left(h^{(1)} \cdot W^{(2)} + b^{(2)}\right) \tag{2.3}$$

Općenito:

$$h^{(i)} = f^{(i)}\left(h^{(i-1)} \cdot W^{(i)} + b^{(i)}\right) \tag{2.4}$$

Oznaka $h^{(i)}$ prikazuje skrivene jedinice i -tog sloja mreže, $W^{(i)}$ je matrica težina sloja te je $b^{(i)}$ prag (engl. *bias).* Funkcija f je aktivacijska funkcija nelinearnosti sloja.



Slika 2.1: Jednostavna umjetna neuronska mreža

Duboke neuronske mreže sadrže mnogo međusobno povezanih neurona, odnosno, veliki broj skrivenih slojeva. Duboke neuronske mreže koje sadrže konvolucijske slojeve nazivamo duboke konvolucijske neuronske mreže i one imaju široku primjenu u području računalnog vida.

2.2 Konvolucijski slojevi

Konvolucijske neuronske mreže osim strukturne informacije imaju za cilj iskoristiti i prostornu informaciju slike [1]. Konvolucijski sloj temelji se na linearnoj transformaciji koja se naziva konvolucija. Podaci na računalu su diskretni stoga se koristi diskretna konvolucija definirana kao:

$$I(i)*K(i)=\sum_{m} I(m)\cdot K(i-m) \tag{2.5}$$

gdje je I ulaz u konvoluciju, a K filtar (engl. *kernel).* Konvolucije u sklopu računalnog vida najviše koristimo za dvodimenzionalne podatke, odnosno slike. Tada je filtar dvodimenzionalan:

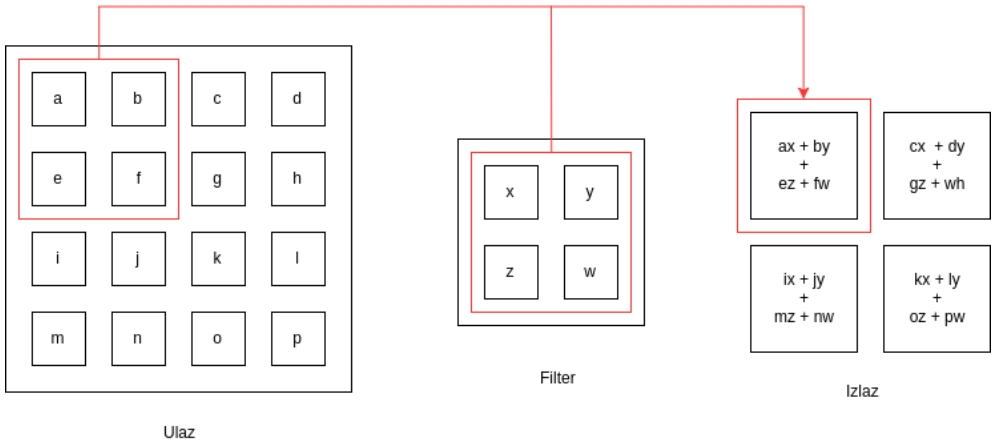
$$I(i,j)*K(i,j)=\sum_{m}\sum_{n} I(m,n)K(i-m,j-n) \tag{2.6}$$

Izlazi dvodimenzionalne konvolucije nazivaju se mape značajki. Postupak provođenja operacije konvolucije nad slikom u sklopu dubokog učenja je sljedeći: unutar dvodimenzionalnog ulaza definira se prozor veličine filtra koji se naziva receptivno polje. Vrijednosti unutar receptivnog polja množe se vrijednostima filtra te se

sumiraju. Filter se primjenjuje po cijeloj ulaznoj slici. U modernim konvolucijskim modelima najčešće su korišteni kvadratni filteri čije su dimenzije neparan broj (npr. 1, 3, 5, ...). Jedan konvolucijski sloj može imati više takvih filtera čime nastaje više mapa značajki.

Parametri koji određuju dimenzije izlazne mape su korak konvolucije S (engl. *Stride*), popunjavanje do rubova P (engl. *Padding*) te veličina filtra k i dimenzije ulaza $m \cdot n$, gdje je m širina, a n visina slike [1]. Korak konvolucije određuje pomak filtera po širini i visini ulazne mape značajki ili ulazne slike, a popunjavanje do rubova koristimo uglavnom kada želimo zadržati rezoluciju nakon konvolucijskog sloja. Izračun dimenzija izlazne mape definiran je sljedećim formulama:

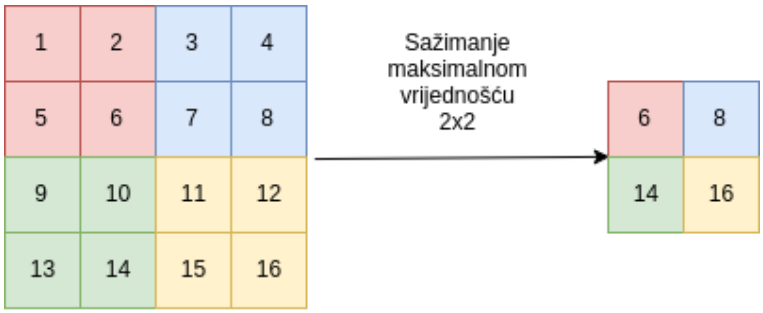
$$w = \frac{m + 2 \cdot P - k}{S} + 1 \qquad h = \frac{n + 2 \cdot P - k}{S} + 1 \tag{2.7}$$



Slika 2.2: Izgled dvodimenzionalne konvolucije uz veličinu filtera k=2, bez popunjavanja do rubova i korakom konvolucije S=2.

Prednost konvolucijskih mreža je puno manji broj težina u odnosu na mreže s potpuno povezanim slojevima. Korištenjem konvolucije spremaju se značajke kao što su rubovi (i slični oblici, ovisno o domeni skupa podataka) koje ne zauzimaju veliki volumen slike. Broj težina trenutnog sloja računa se formulom $k \cdot k \cdot r \cdot f$, gdje je r dubina slike (npr. tri kanala RGB slike), a f broj značajki trenutnog sloja.

Još jedna od komponenti konvolucijskog sloja je sloj sažimanja (engl. *pooling layer*). To je vrsta poduzorkovanja podataka. Uzima se prozor veličine $k \times k$, gdje je k konstanta sažimanja, odnosno faktor za koji se ulazne mape smanjuju.



Slika 2.3: Sažimanje maksimalnom vrijednošću

Kod sažimanja maksimalnom vrijednošću (engl. *max pooling*), u svakom dijelu operacije uzimamo maksimalnu vrijednost, a kod sažimanja srednjom vrijednošću (engl. *average pooling*) uzimamo srednju vrijednost unutar prozora. Postupak sažimanja ostvaruje invarijantnost na lokalne translacije te može biti koristan kada nam je bitnije da neka značajka postoji nego poznavanje njene točne lokacije na ulaznom podatku [1]. Sličan efekt može se postići korištenjem konvolucije s većim korakom.

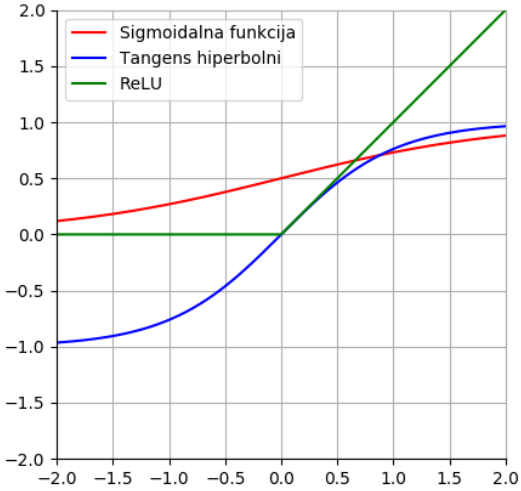
2.3 Aktivacijske funkcije

Aktivacijske funkcije služe kako bi se postigla nelinearnost između slojeva neuronske mreže jer bez njih mreža ne bi mogla naučiti nelinearnost među podacima. Najčešće korištene aktivacijske funkcije su: sigmoidalna (slika 2.8), tangens hiperbolni (2.9) i ReLU (2.10). One se nalaze nakon izlaza konvolucijskog sloja mreže. Grafovi tih funkcija nalaze se na slici (2.4).

$$f(x) = \sigma(x) = \frac{1}{1+e^{(-x)}} \tag{2.8}$$

$$f(x) = \tanh(x) = \frac{1-e^{(-2x)}}{1+e^{(-2x)}} \tag{2.9}$$

$$f(x) = max(0, x) \tag{2.10}$$



Slika 2.4: Grafovi aktivacijskih funckija

Sigmoidalna funkcija je česta aktivacijska funkcija koja monotono raste te se njen izlaz asimptotski približava nekoj konačnoj vrijednosti (1 ili -1), dok ulaz te aktivacije raste prema ±∞. Ona ima nekoliko nedostataka: kada uđe u zasićenje, pojavljuje se problem nestajućeg gradijenta (eng. *vanishing gradient*), izlazi nisu centrirani oko 0 te je računanje eksponenta skupa operacija. Tangens hiperbolni obično se ponaša bolje od sigmoide jer sliči identitetu u dijelu oko $x=0$, što omogućuje jednostavan transfer gradijenata unatrag. Također, izlazi su centrirani oko nule, ali još uvijek dolazi do problema nestajućeg gradijenta u zasićenju (kada su uzlazne vrijednosti jako visoke u pozitivnom ili negativnom smjeru). Aktivacijska funkcija ReLU (engl. *Rectified Linear Unit*) se vrlo često koristi u dubokim mrežama jer modeli puno brže konvergiraju. U pozitivnom dijelu $x>0$ nema zasićenja što za posljedicu ima smanjen problem nestajućeg gradijenta. Izračun ReLU je vrlo efikasan, ali u prolazu prema naprijed, ako je $x<0$ neuron postaje neaktivan te gasi gradijent prilikom prolaska unatrag.

2.4 Optimizacijski postupak

Optimizacijski postupak pronalazi parametre modela za koje je odabrana funkcija gubitka minimalna. Cilj je minimizirati očekivanu pogrešku generalizacije (rizik) danu s (2.10). Pošto distribucija podataka nije poznata, problem se svodi na optimizacijsku procjenu prave distribucije na temelju empirijske distribucije određene uzorcima za učenje u nadi da će smanjenjem empirijskog rizika padati i očekivani gubitak [2].

$$\mathrm{E}_{(\vec{x},\vec{y})aprox.\,\hat{p_{data}}}[L(f(\vec{x},\theta),\vec{y})]=\frac{1}{n}\sum_{i=1}^{n} L(f(\vec{x^{(i)}};\theta),\vec{y^{(i)}}) \tag{2.10}$$

Nekada nije moguće ili praktično minimizirati pravi gubitak jer često nije kontinuiran i/ili derivabilan. Rješenje je optimirati nadomjesnu funkciju gubitka (engl. *surrogate loss function*) koja ima povoljnija svojstva i možemo ju poopćiti na sljedeći način :

$$\hat{\theta}=\min_{\theta}\frac{1}{n}\sum_{i=1}^{n} L_s(f(\vec{x^{(i)}},\theta),\vec{y^{(i)}}) \tag{2.11}$$

Metoda učenja širenjem unazad (engl. *backpropagation*) je osnovni optimizacijski postupak učenja dubokih mreža. Postupak optimizacije se obavlja do zadovoljavanja zadanih kriterija konvergencije ili do isteka nekog postavljenog uvjeta. Optimizacijski postupci se dijele na:

- postupke koji koriste čitav skup primjera za učenje (engl. *batch*),
- postupke koji koriste jedan po jedan primjer iz skupa za učenje ili stohastički (engl. *stohastic*),
- postupke koji koriste podskup primjera u koraku ili mini-grupe (engl. *mini-batch*).

Pri optimizaciji modela dubokog učenja najviše se koristi učenje nad mini-grupama jer brže vodi do konvergencije. Algoritmi učenja nad mini-grupama računaju gradijent na dijelu podataka, što znači da aproksimiraju gradijent cijelog skupa podataka. Tijekom aproksimacije unose dodatni šum, no upravo je šum koristan pri ne-konveksnoj optimizaciji jer može izvući postupak iz lokalnog minimuma ili sedlastih područja u kojima je zapeo.

Gradijentni spust (engl. *gradient descent*) je iterativan način minimizacije funkcije gubitka $L(\theta)$ određene parametrima modela θ . To čini ažuriranjem parametara θ u suprotnom smjeru gradijenata $L(\theta)$. Formalno:

$$\theta^{(i+1)}=\theta^{(i)}-\eta\nabla_{\theta}L(x^{(i)},y^{(i)};\theta) \tag{2.12}$$

gdje η predstavlja stopu učenja. Stopa učenja određuje veličinu koraka koji koristimo da bismo došli do (lokalnog) minimuma. Definiranje stope učenja vrlo je bitno jer ima značajan utjecaj na učinak modela. Bira se eksperimentalno na temelju promatranja krivulje učenja koja prikazuje vrijednost funkcije gubitka ovisno o vremenu.

2.5 Normalizacija po grupama

Normalizacija po grupama (engl. *batch normalization*) je adaptivna reparametrizacija podataka tijekom procesa treniranja duboke mreže. Pomak kovarijance (engl. *covariate shift*) odnosi se na promjenu raspodjele ulaznih vrijednosti modela. Pomak kovarijance može biti problem jer se ponašanje modela može promjeniti kada se mijenja distribucija ulaznih podataka. Osnovna ideja normalizacije po grupama jest ograničavanje pomaka kovarijance normalizacijom aktivacija svakoga sloja, odnosno transformiranje vrijednosti podataka u podatke sa srednjom vrijednošću nula i varijancom jedan [1]. Gradijent govori kako ažurirati svaki parametar, pod pretpostavkom da se drugi slojevi ne mijenjaju. Slojeve ažuriramo istovremeno i to stvara problem jer je izlaz jednog sloja ulaz u sljedeći. Normalizacija po grupama pruža elegantan način ublažavanja navedenih problema. Posljedica je da grupe tijekom treniranja imaju sličnu distribuciju te se postiže bolje propagiranje gradijenata, a time i brže učenje mreže.

Neka je $\{x^{(i)},x^{(n)}\}$ minigrupa veličine n te β,γ parametri koji se uče. Izlaz je normalizirana mini-grupa $\{y^{(i)},y^{(n)}\}$ veličine n koja se računa kao:

$$\mu_{\beta}=\frac{1}{m}\sum_{i=1}^{n}x_i \tag{2.13}$$

$$\sigma_{\beta}^{2}=\frac{1}{m}\sum_{i=1}^{n}(x_i-\mu_{\beta})^2 \tag{2.14}$$

$$\hat{x}_i = \frac{x_i - \mu_\beta}{\sqrt{\sigma_\beta^2 + \eta}} \tag{2.15}$$

$$y_i = \gamma \hat{x}_i + \beta \tag{2.16}$$

Normalizacija svakog ulaza sloja može promijeniti neke bitne informacije o značajkama stoga je zadnji korak normalizacije grupe transformacija koja omogućava da se originalne vrijednosti mogu vratiti iz normaliziranih ulaznih podataka. Parametri te transformacije β, γ se uče kao i ostali parametri modela. Normalizacija po grupama osim pomoći u treniranju, vrši regularizaciju dubokog modela što spriječava prenaučenost modela. U konvolucijskim slojevima je bitno primjetiti da se zajednička sredina i varijanca računaju za svaku izlaznu mapu značajki [3].

2.6 Prijenos znanja

U praksi, mali je postotak ljudi koji trenira cijelu neuronsku mrežu od nule zbog nedostatka količine podataka ili resursa za treniranje. Umjesto toga, uobičajena su tri scenarija:

- neuronska mreža kao fiksni ekstraktor značajki - uzimanje značajki izlaza konvolucijskog sloja prethodno istrenirane mreže u svrhu klasifikacije na drugačijem skupu podataka, drugom broju klasa itd.,
- fino podešavanje – dotreniravanje prethodno istrenirane mreže nad skupom podataka koji prethodno nije viđen, ali je "domena podataka" slična,
- prethodno istrenirani modeli – korištenje već gotove ili fino podešene mreže.

Pri odlučivanju koji tip prijenosa znanja treba iskoristiti, u obzir se mora uzeti veličina i sličnost novog skupa podataka u odnosu na onaj na kojem je trenirana mreža.

2.7 Arhitektura ResNet

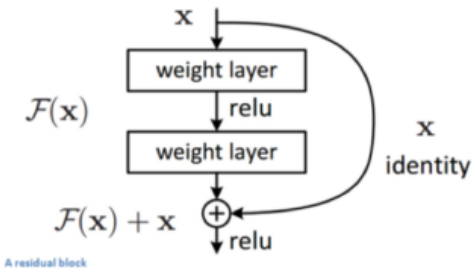
S obzirom na ulazni podatak x cilj neuronske mreže je pronaći korisnu funkciju mapiranja $H(x)$. Funkcija $H(x)$ (u sklopu klasifikacije) pokušava mapirati ulaz na izlaz, odnosno klasificirati dani podatak u jednu od specifičnih klasa. Neka je:

$$F(x)=H(x)-x \tag{2.17}$$

gdje je $F(x)$ rezidualna funkcija (engl. *residual function*). Hipoteza autora [4] arhitekture ResNet je da je lakše optimizirati rezidualnu funkciju $F(x)$ nego originalnu funkciju mapiranja $H(x)$. Izvorna funkcija mapiranja $H(x)$ je ono što je potrebno pa se u rezidualnim mrežama koristi:

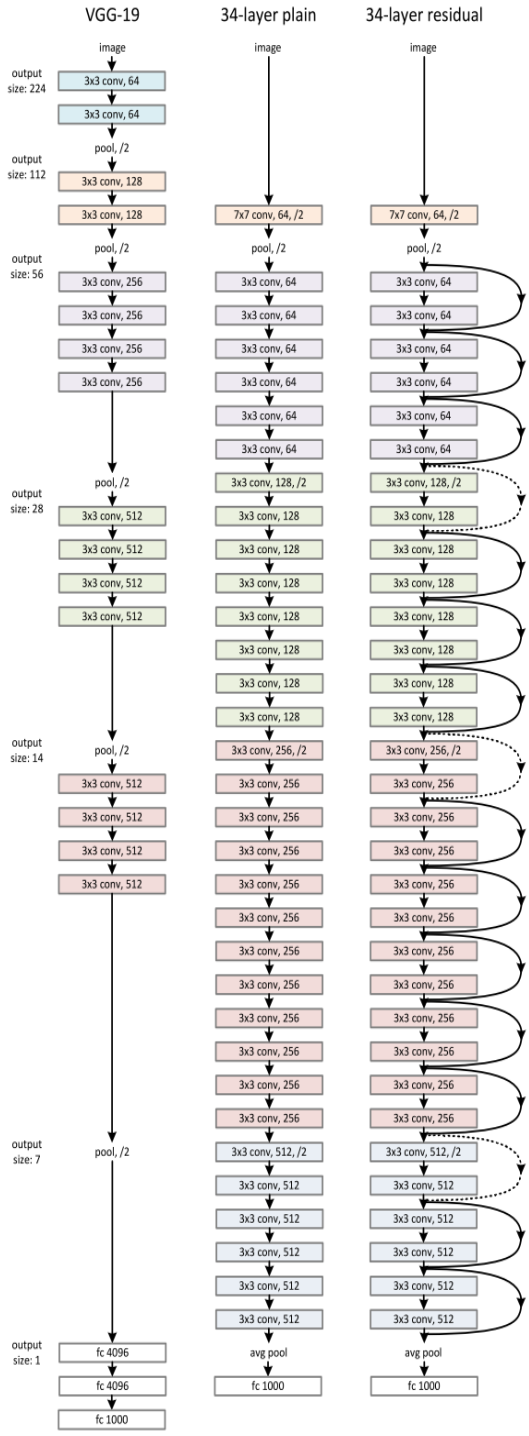
$$H(x)=F(x)+x \tag{2.18}$$

Osnovna gradivna jedinica ResNet arhitekture je ResNet blok prikazan na slici (2.5) gdje su veze preko slojeva težina zvane preskočne veze (engl. *skip connections*).



Slika 2.5: Rezidualni blok. Slika preuzeta iz [4].

Na slici (2.6) nalazi se usporedba VGG-19 mreže, obične konvolucijske mreže sa 34 sloja te mreže ResNet arhitekture sa 34 sloja.



Slika 2.6: Usporedba “tradicionalne” konvolucijske arhitekture sa ResNet arhitekturom. Slika preuzeta iz [4].

2.8 Potpuno konvolucijske mreže

Potpuno konvolucijske mreže (engl. *Fully Convolutional Networks* ili *FCN*) su neuronske mreže koje se sastoje samo od konvolucijskih slojeva uz opcionalne slojeve sažimanja. U praksi to znači da je zadnji potpuno povezani sloj zamijenjen konvolucijskim slojem. Veoma često se mreže namijenjene za klasifikaciju pretvaraju u potpuno konvolucijske mreže. To se može napraviti tako da se uzme konvolucijska mreža koja svojim rezultatima, domenom podataka na kojoj je učena i sl. odgovara problemu i zatim se svi potpuno povezani slojevi pretvore u konvolucijske slojeve veličine 1×1 . Dobro svojstvo koje se dobiva je mogućnost dovođenja slika proizvoljne veličine na ulaz mreže.

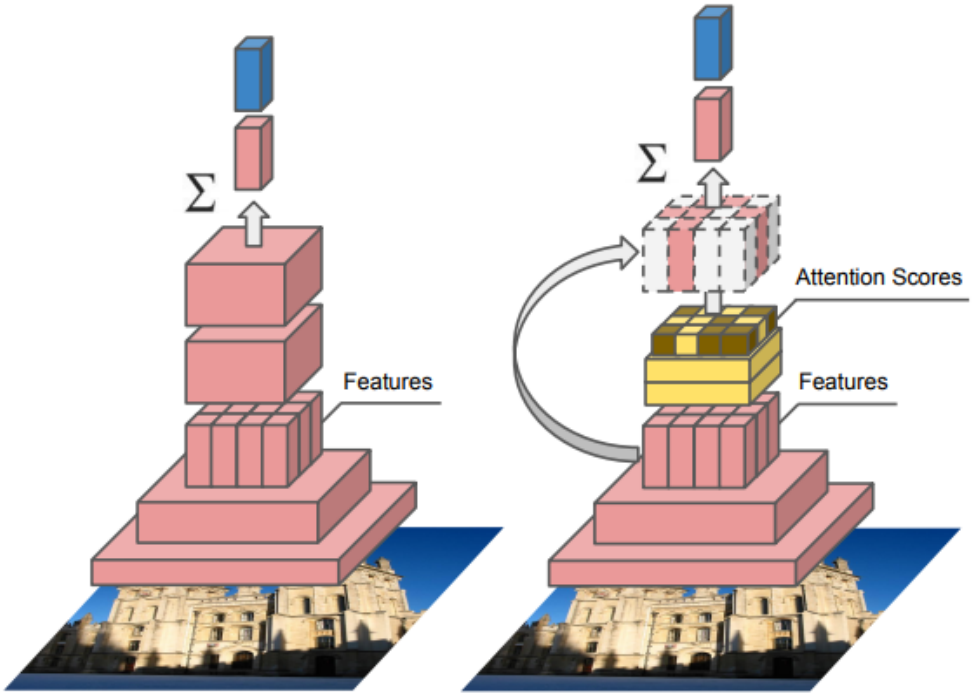
U ovom je radu korištena potpuno konvolucijska neuronska mreža Resnet50 čija je ideja opisana u prethodnom poglavlju.

3. Duboke lokalne značajke

Pretraživanje slikovnih baza prema slikovnom sadržaju temeljni je zadatak računalnog vida budući da je povezano s raznim praktičnim primjenama poput detekcije objekata, vizualnog prepoznavanja mjesta, prepoznavanja proizvoda... Unatoč najnovijim dostignućima u globalnim deskriptorima temeljenim na konvolucijskim neuronskim mrežama [5], njihovu učinkovitost lako može omesti širok raspon izazovnih uvjeta poput okluzija, varijacija u pogledu i osvjetljenju. Globalni deskriptori nemaju mogućnost pronalaženja podudaranja slika na razini dijelova slike (engl. *patch level matches*). Kao rezultat, teško je dohvatiti slike na temelju djelomičnog podudaranja u prisustvu izazovnih uvjeta. U novijem trendu, lokalne značajke konvolucijskih neuronskih mreža predložene su za podudaranje na razini dijelova slike. Međutim, takve tehnike nisu optimizirane specifično za pretraživanje slikovnih baza prema slikovnom sadržaju jer im nedostaje sposobnost otkrivanja semantički smislenih značajki [5] i pokazuju ograničenu točnost.

U ovom poglavlju prikazana je vrsta konvolucijske neuronske mreže zvana DELF (DEep Local Features) [5] kao novi deskriptor značajki (slika 2.5). Model je baziran na ekstrakciji konvolucijskih značajki s pažnjom (engl. *attention*). Pažnja je mjera kojom se eksplicitno određuje relevantnost pojedinih značajki. DELF je podijeljen u dva dijela: fino podešena mreža ResNet50 i model pažnje (engl. *attention model*). Model pažnje implementiran je kao dvoslojna konvolucijska neuronska mreža. DELF je treniran slabim nadzorom (engl. *weak supervision*) koristeći oznake klasa na razini slike.

Ovakvim pristupom model pozornosti čvrsto je povezan sa značajkama; ponovno koristi istu arhitekturu konvolucijske neuronske mreže i generira mjeru važnosti značajke pomoću vrlo malo dodatnih računanja. To omogućuje ekstrakciju deskriptora (značajki) i ključnih točaka jednim prolazom slike kroz mrežu.



Slika 3.1: Lijevo - fino podešavanje deskriptora, Desno - treniranje modela pažnje. U oba slučaja se koristi gubitak unakrsne entropije. Slika preuzeta s [5].

3.1 Ekstrakcija gustih značajki

Općenito, guste značajke ekstrahiraju se koristeći izlaz jednog od konvolucijskih slojeva neuronske mreže trenirane klasifikacijskim gubitkom.

U ovom radu se koristi potpuno konvolucijska neuronska mreža ResNet50 [4] prethodno trenirana na skupu podataka ImageNet [6]. Mreža je fino podešena na skupu podataka The Landmarks [7] (skup podataka znamenitosti koji se sastoji od 586 različitih klasa) te se kao izvor značajki koristi izlaz iz *conv4_x* (Tablica 1.) konvolucijskog bloka. Pretpostavka je da bi značajke sljedećeg konvolucijskog bloka (*conv5_x*) mogle bi biti previše specifične, a značajke prethodnog konvolucijskog bloka (*conv3_x*) nedovoljno specifične za pretraživanje slikovnih baza prema slikovnom sadržaju. Dobivene mape značajki smatraju se gustom mrežom lokalnih deskriptora. Značajke su lokalizirane na temelju svojih receptivnih polja koja se mogu izračunati uzevši u obzir konfiguraciju konvolucijskih slojeva i slojeva sažimanja korištene mreže. Točna lokacija značajke na slici uzima se kao centar receptivnog polja.

Ime sloja	Veličina izlaza	ResNet50
conv1	112×112	$7\times7, 64$, *korak* 2
conv2_x	56×56	3×3 *sažimanje maksimalnom vrijednošću, korak* 2
		$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$
conv4_x	14×14	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$
conv5_x	7×7	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	*Sažimanje usrednjavanjem,* 1000 *dim potpuno povezani sloj, softmax*

Tablica 1. Arhitektura modela ResNet50 [4] za ulazne slike dimenzija 224x224. Smanjenje dimenzija događa se nakon conv3_1, conv4_1 i conv5_1 sa korakom 2

3.2 Odabir ključnih značajki temeljen na pozornosti

Umjesto izravne uporabe ekstrahiranih značajki pri pretraživanju slikovnih baza prema slikovnom sadržaju, u ovom se radu koristi tehnika odabira podskupa značajki. Budući da je značajan dio gusto ekstrahiranih značajki nerelevantan za problem pretraživanja slikovnih baza prema slikovnom sadržaju, odabir ključnih značajki (engl. *keypoint selection*) važan je za točnost i učinkovitost sustava.

3.2.1 Učenje pažnje

Pažnja (engl. *attention*) je mjera kojom se eksplicitno određuje relevantnost pojedinih značajki. Formalno: neka je $f_n \in R^d, n=1,\ldots,N$ d - dimenzionalna značajka. Cilj je naučiti funkciju mjere $\alpha(f_n;\theta)$ (engl. *score function*) za svaku značajku, gdje su θ parametri funkcije $\alpha(.)$. Izlaz mreže je onda dan kao:

$$y = W\left(\sum_n \alpha(f_n;\theta)\cdot f_n\right) \tag{2.19}$$

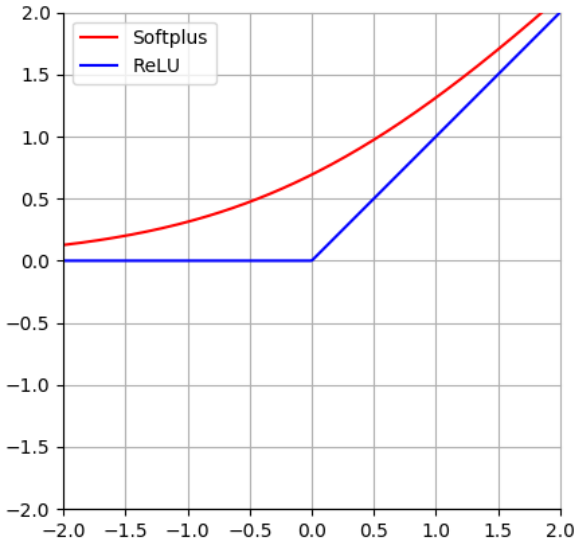
gdje je $W \in R^{M\times d}$ skup težina posljednjeg potpuno povezanog sloja konvolucijske neuronske mreže treniranih na M klasa.

Parametri funkcije mjere, uz gubitak unakrsne entropije, treniraju se gradijentnim spustom na sljedeći način:

$$\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial y}\sum_n \frac{\partial y}{\partial \alpha_n}\frac{\partial \alpha_n}{\partial \theta} = \frac{\partial L}{\partial y}\sum_n W f_n \frac{\partial \alpha_n}{\partial \theta} \tag{2.20}$$

gdje se gradijent funkcije mjere $\alpha_n \equiv \alpha(f_n;\theta)$ s obzirom na parametre θ računa jednako kao i u standardnom višeslojnom perceptronu.

Funkciju mjere $\alpha(.)$ definiramo kao ne-negativnu da bi spriječili učenje negativnih težina. Dizajnirana je kao dvoslojna konvolucijska mreža sa softplus [8] (glatka aproksimacija funkcije ReLU; $L(x)=\log(1+e^x)$) aktivacijskom funkcijom (slika 3.2) te konvolucijskim filterima veličine (1×1) .



Slika 3.2: Usporedba aktivacijskih funkcija softplus i ReLU

3.2.2 Karakteristike

Jedan od nekonvencionalnih aspekata prethodno opisanog modela je da se selekcija ključnih točaka odvija nakon ekstrakcije deskriptora za razliku od postojećih tehnika poput SIFT-a [9] ili LIFT-a [10] gdje su ključne točke prvo ekstrahirane i nakon toga opisane. Tradicionalni detektori ključnih točaka usredotočeni su na otkrivanje ključnih točaka pod različitim uvjetima snimanja temeljenih samo na njihovim osnovnim karakteristikama (engl. *low level characteristics*). Međutim, za teške probleme pretraživanja slikovnih baza prema slikovnom sadržaju, kritično je odabiranje ključnih točaka koje vrše razlike objekata na razini instance. Prethodno opisana arhitektura postiže oba cilja treniranjem modela koji kodira sliku u mapu značajki i uči kako i koje značajke odabrati.

4. Implementacija

U okviru rada implementiran je model opisan u prethodnom poglavlju. Za cijelu programsku izvedbu korišten je programski jezik Python 3, razvojni okvir TensorFlow [11] i osnovne knjižice za obradu i manipulaciju korištenim podacima poput Numpy-a [12], matplotlib-a [13], OpenCV-a [14]... TensorFlow je razvojni okvir otvorenog koda koji u prvom koraku generira računski graf, a nakon toga obavlja izračun računskog grafa. Korištena je i knjižica TF-Slim [15] koja služi za definiranje, treniranje i procjenu složenih modela. Komponente TF-Slima mogu se slobodno miješati s nativnim TensorFlowom (kao i drugim okvirima definiranim u tensorflow.contrib-u).

Eksperimentalni rezultati provedeni su na serverskom računalu s operacijskim sustavom Linux Ubuntu koje ima Nvidia grafičku karticu GeForce GTX 1080 (8 GB).

5. Rezultati

U ovom poglavlju navedeni su podatkovni skupovi korišteni za treniranje i evaluaciju. Nadalje, navode se metode evaluacije i dobiveni rezultati.

5.1 Podatkovni skupovi

5.1.1 ImageNet

ImageNet [6] je baza podataka slika organizirana prema hijerarhiji WordNet-a [16] u kojima je svaki čvor hijerarhije prikazan stotinama ili tisućama slika. Trenutno je preko 14 miljuna URL-ova slika ručno anotirano na 1000 klasa. Model ResNet50 [4] treniran je na ovom skupu podataka.

Slika 5.1: Skup podataka ImageNet

5.1.2 Skup podataka The Landmarks

Skup podataka The Landmarks (u radu zvan kao skup podataka znamenitosti) [7] podijeljen je u dva dijela: potpuna verzija (engl. *full version*; u radu zvana kao LF) koja se sastoji od 118 112 slika i čista verzija (engl. *clean version*; u radu zvana kao LC) koja se sastoji od 29535 slika podijeljenih kroz 586 klasa gdje je LC podskup od LF. Skup podataka znamenitosti sadrži veliku varijabilnost unutar klasa (diverzitet kuta ili skale pogleda i/ili osvijetljenja, slike interijera znamenitosti pa čak i nezanemarive količine slika koje ne pripadaju promatranim klasama) što ne predstavlja problem pri klasifikaciji (regularizacijski efekt šuma), ali može biti problem pri treniranju mreže za podudaranje slika na razini instance (engl. *instance-level matching*). Zbog toga nastaje potreba "čišćenja" skupa podataka. Skup podataka LF pretvoren je u skup podataka LC na sljedeći način:

1. uzimaju se sve slike koje pridapaju pojedinoj klasi,
2. za svaki par slika pronalazi se broj zajedničkih točaka (metoda SIFT i geometrijska verifikacija [7]),
3. konstruira se graf čiji su čvorovi slike, a bridovi vrijednosti dobivene iz koraka 2.,
4. odstranjuju se bridovi s niskim vrijednostima i ekstrahira se samo najveća povezana komponente grafa
5. postupak se ponavlja za svaku klasu zasebno

Skup podataka LC korišten je za fino podešavanje mreže ResNet50, a skup podataka LF korišten je za učenje modela pažnje. Skup podataka znamenitosti sastoji se i od dva validacijska skupa (21326 i 4369 slika) gdje je manji podskup većega i korišten je kao skup za validaciju pri učenju. Oboje sadrže isti broj klasa kao i skupovi za učenje. Sa interneta je preuzet samo podskup slika zbog nepravilnih URL-ova te se brojevi ne podudaraju u potpunosti s radovima [7] [17] .

Slika 5.2: Slučajno odabranih 12 slika iz skupa podataka znamenitosti

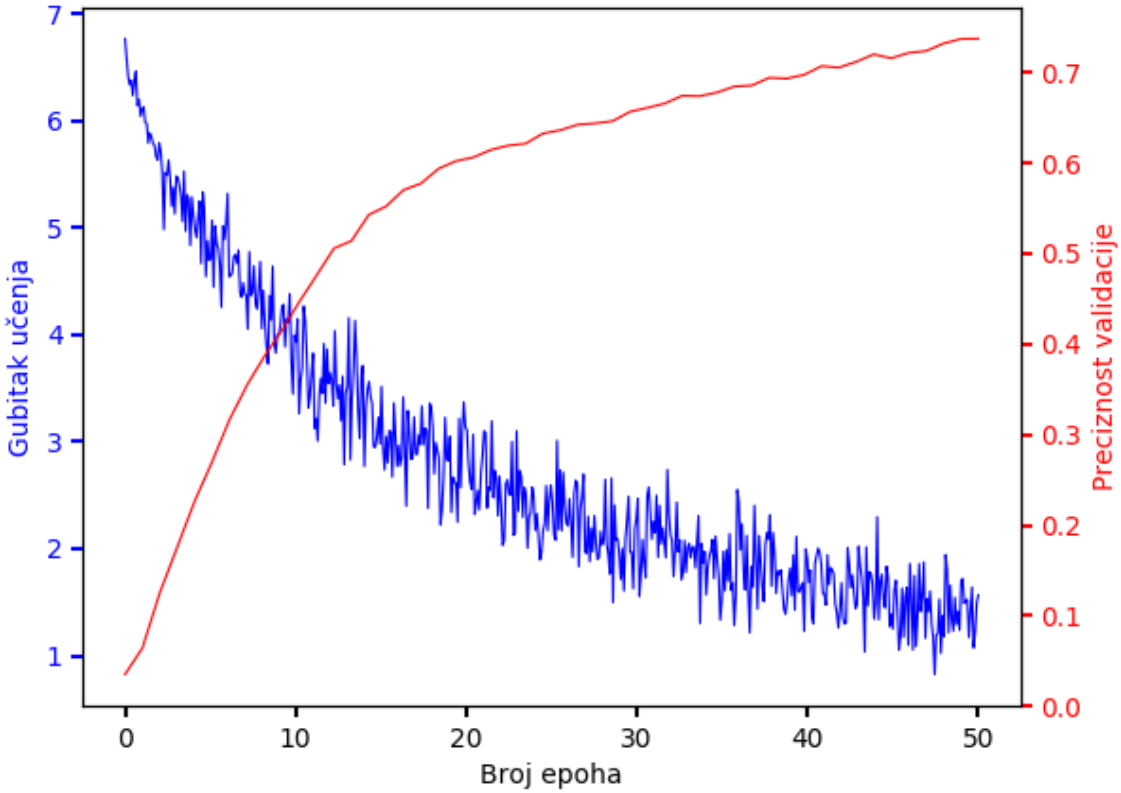
5.1.3 Oxford5k

Skup podataka Oxford5k [18] sastoji se od 5062 slike skupljenih sa Flickr-a za upite vezane uz pojedine znamenitosti grada Oxforda. Skup se sastoji od 55 različitih upita, 5 po svakoj od 11 klasa, preko kojih možemo testirati sustav pretraživanja slikovnih baza prema slikovnom sadržaju. Svakom upitu pripadaju tri vrste klasa slika: *good* (lijepe, jasne slike objekata), *OK* (više od 25% objekta je prisutno na slici) i *junk* (manje od 25% objekta je prisutno na slici, puno šuma, distorzija). Sve tri klase korištene su u eksperimentima.

Slika 5.3: Primjer slika iz podatkovnog skupa Oxford5k

5.2 Treniranje

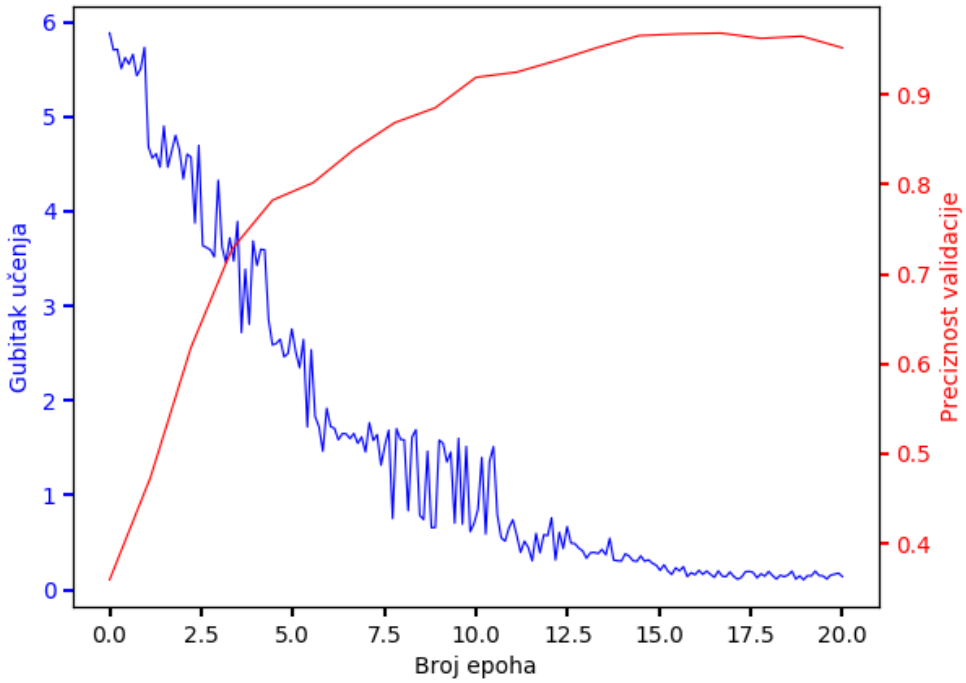
Kao što je navedeno u 3. poglavlju, treniranje je podijeljeno u 2 dijela: fino podešavanje modela ResNet50 te nakon toga treniranje modela pažnje. Mreža ResNet50 fino je podešena na skupu podataka LC opisanog u 5.1.2 koji sadrži 586 različitih klasa znamenitosti. Svaka slika je centralno podrezana i skalirana na 250×250 te je nakon toga uzet slučajni dio dimenzija 224×224 koji se dalje koristi za treniranje. Korištena je veličina mini-grupe od 64 slike te je ResNet50 fino podešavan 50 epoha uz fiksnu stopu učenja od 0.001 koristeći gradijentni spust kao optimizacijski postupak. Postupak finog podešavanja trajao je oko 10 sati. Na slici 5.4 prikazan je graf procesa finog podešavanja: x os predstavlja broj epoha, na lijevoj y osi prikazan je gubitak na skupu za treniranje, a na desnoj y osi prikazana je preciznost na validacijskom skupu opisanog u 5.1.2.



Slika 5.4: Krivulje gubitka učenja i preciznosti validacije pri finom podešavanju mreže ResNet50

Na slici 5.3 može se vidjeti kako preciznost na validacijskom skupu dostiže već poprilično dobar rezultat od 73%. Nakon pedesete epohe gubitak učenja i dalje pada, ali preciznost na validacijskom skupu prvo stagnira da bi nakon desetak epoha počela padati što znači da se mreža počinje prenaučavati (engl. *overfitting*).

Iako bi se oba modela mogla trenirati zajedno, takav način treniranja stvara modele slabijih performansi [5]. Pri treniranju modela pažnje, korišten je skup podataka LF opisan u 5.1.2 koji sadrži 586 klasa. Svaka slika je centralno podrezana i skalirana na 900×900 te je nakon toga uzet slučajni dio dimenzija 720×720 . Korištena je veličina mini-grupe od 64 slike te je model pažnje učen 20 epoha uz visoku stopu učenja s inicijalnom vrijednošću 1 i eksponencijalnim opadanjem vrijednosti (engl. *exponential decay*). Korišten je gradijensti spust kao optimizacijski postupak. Postupak treniranja pažnje trajao je oko 20 sati. Na slici 5.5 prikazan je graf procesa učenja modela pažnje: x os predstavlja broj epoha, na lijevoj y osi prikazan je gubitak na skupu za treniranje, a na desnoj y osi prikazana je preciznost na validacijskom skupu opisanog u 5.1.2.



Slika 5.5: Krivulje gubitka učenja i preciznosti validacije pri učenju modela pažnje

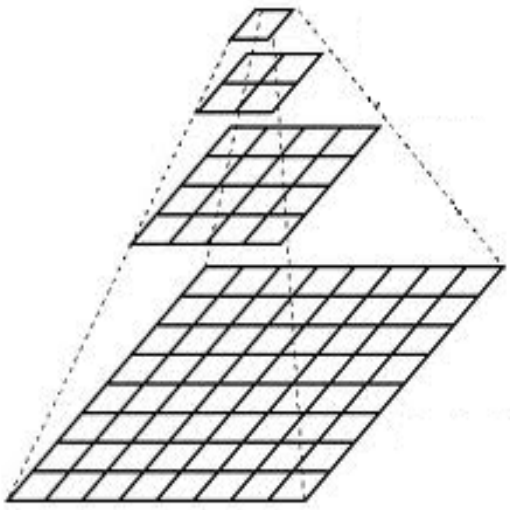
Eksperimentima se pokazalo da model zapinje u lokalnom minimumu pri početku učenja. Da bi model "preskočio" lokalni minimum, inicijalno je korištena visoka stopa učenja u vrijednosti od 1 s eksponencijalnim opadanjem vrijednosti.

5.3 Testiranje

Cilj klasifikacije slika je predviđanje kategorije kojoj promatrana slika pripada dok je cilj pretraživanja slikovnih baza prema slikovnom sadržaju pronalazak "najbliže" slike iz baze uz danu sliku upita. Mjera bliskosti slika proizvoljno se definira i ovisi o domeni problema i načinu implementacije sustava za pretraživanje slikovnih baza prema slikovnom sadržaju (npr. dvije slike pripadaju istoj klasi ako im je vrijednost kosinus udaljenosti vektorskih reprezentacija dovoljno mala).

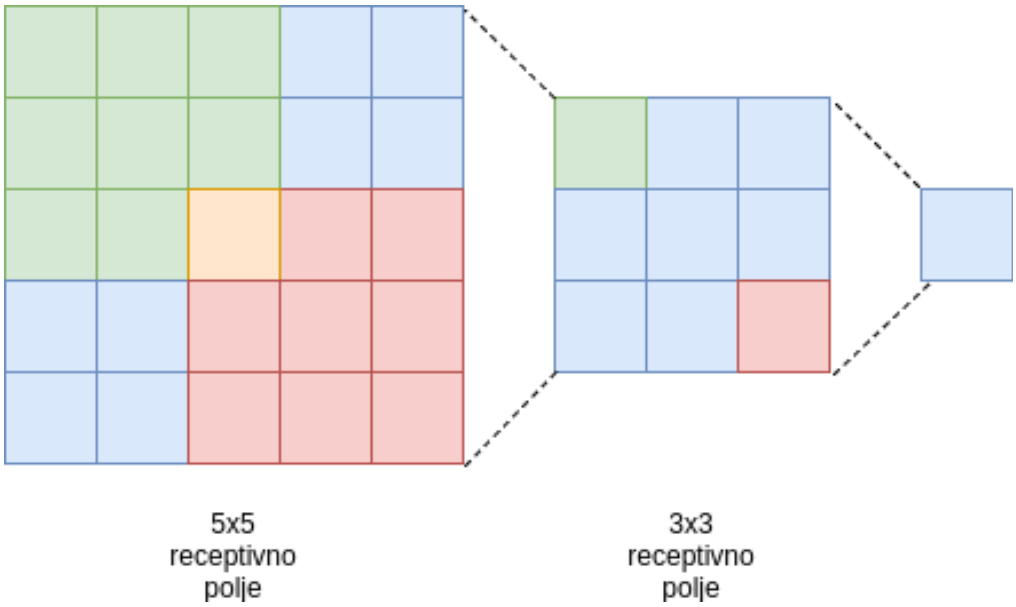
5.3.1 Ekstrakcija značajki

Povoljno svojstvo potpune konvolucijske arhitekture je da slike na ulaz u mrežu mogu doći u varijabilnim veličinama. Zato se za ekstrakciju značajki za svaku sliku konstruira slikovna piramida (slika 5.6) koristeći skale međusobno udaljene $\sqrt{(2)}$ u rasponu od 0.25 do 2.0 . Ovim postupkom dobivamo značajke koje opisuju regije slika različitih veličina. Dodatno, veličina receptivnog polja obrnuto je proporcionalna korištenoj skali.



Slika 5.6: Primjer piramide slike

Svakoj značajki pridružena je vrijednost dobivena iz modela pažnje te se tako odabire n najrelevantnijih silazno sortiranih značajki. Poznavajući arhitekturu mreže, odnosno konfiguraciju konvolucijskih slojeva i slojeva sažimanja te promjene skale promatrane slike, moguće je izračunati veličinu i poziciju receptivnog polja svake značajke na promatranoj slici. Upravo su centri tih receptivnih polja lokalizirane ekstrahirane značajke.



Slika 5.7: Primjer receptivnog polja

5.3.2 Evaluacija

Sustavi za pretraživanje slikovnih baza obično su ocijenjeni na temelju srednje prosječne preciznosti (mAP) koja se izračunava sortiranjem slika u silaznom redoslijedu relevantnosti po upitu i usrednjavanjem srednje preciznosti po pojedinim upitima za različite pragove. Srednja preciznost izračunava se na sljedeći način:

$$P = \frac{TP}{TP + FP} \tag{5.1}$$

$$R = \frac{TP}{TP + FN} \tag{5.2}$$

$$AP = \sum_{t} \left(R_t - R_{t-1} \right) P_t \tag{5.3}$$

gdje je TP broj pogođenih pozitivnih primjera, FP broj pozitivno promašenih primjera, FN negativno promašenih primjera, P je preciznost i R je odziv. Srednja preciznost ili AP je površina ispod krivulje preciznost-odaziv koju konstruiramo rasponom pragova t pri evaluaciji modela. Srednja prosječna preciznost (engl. *medium average precision)* ili mAP je usrednjena srednja preciznost po svim klasama/upitima modela.

U ovome se radu se koristi modificirana verzija preciznosti PRE i odziva REC :

$$PRE = \frac{\left|\Phi_q^{TP}\right|}{\left|\Phi_q\right|} \tag{5.4}$$

$$REC = \left|\Phi_q^{TP}\right| \tag{5.5}$$

gdje je Φ_q skup slika vraćenih od sustava za pretraživanje slikovnih baza za upit q sa danim pragom, a $\Phi_q^{TP} \subseteq \Phi_q$ skup pogođenih pozitivnih primjera. Odziv se uzima kao apsolutna vrijednost pogođenih primjera jer je zbroj u nazivniku izvorne formulacije odziva ($TP+FN$) konstanta.

5.3.3 Rezultati

Pri usporedbi dvaju slika, nad ekstrahiranim značajkama koristi se proizvoljna metoda najbližih susjeda. Uparuju se lokalne značajke čija je međusobna udaljenost manja od hiperparametra K . Točke kojima te značajke pripadaju dodatno prolaze kroz geometrijsku verifikaciju koristeći algoritam RANSAC [19]. RANSAC kao rezultat daje broj uparenih točaka što je mjera sličnosti slika.

Evaluacija je provedena nad skupom podataka Oxford5k. Slike iz skupa podataka Oxford5k nisu korištene pri treniranju. Kao što je opisano u 5.1.3, skup podataka je organiziran na način da ga je povoljno koristiti za evaluaciju sustava za pretraživanje slikovnih baza. Slike vezane za pojedini upit silazno su sortirane po broju zajedničkih točaka (metoda najbližih susjeda i algoritam RANSAC) te je korištena mAP metoda opisana u 5.3.2.

Pri usporedbi se koristi i sljedeće najsuvremenije (engl. *state of the art*) metode:

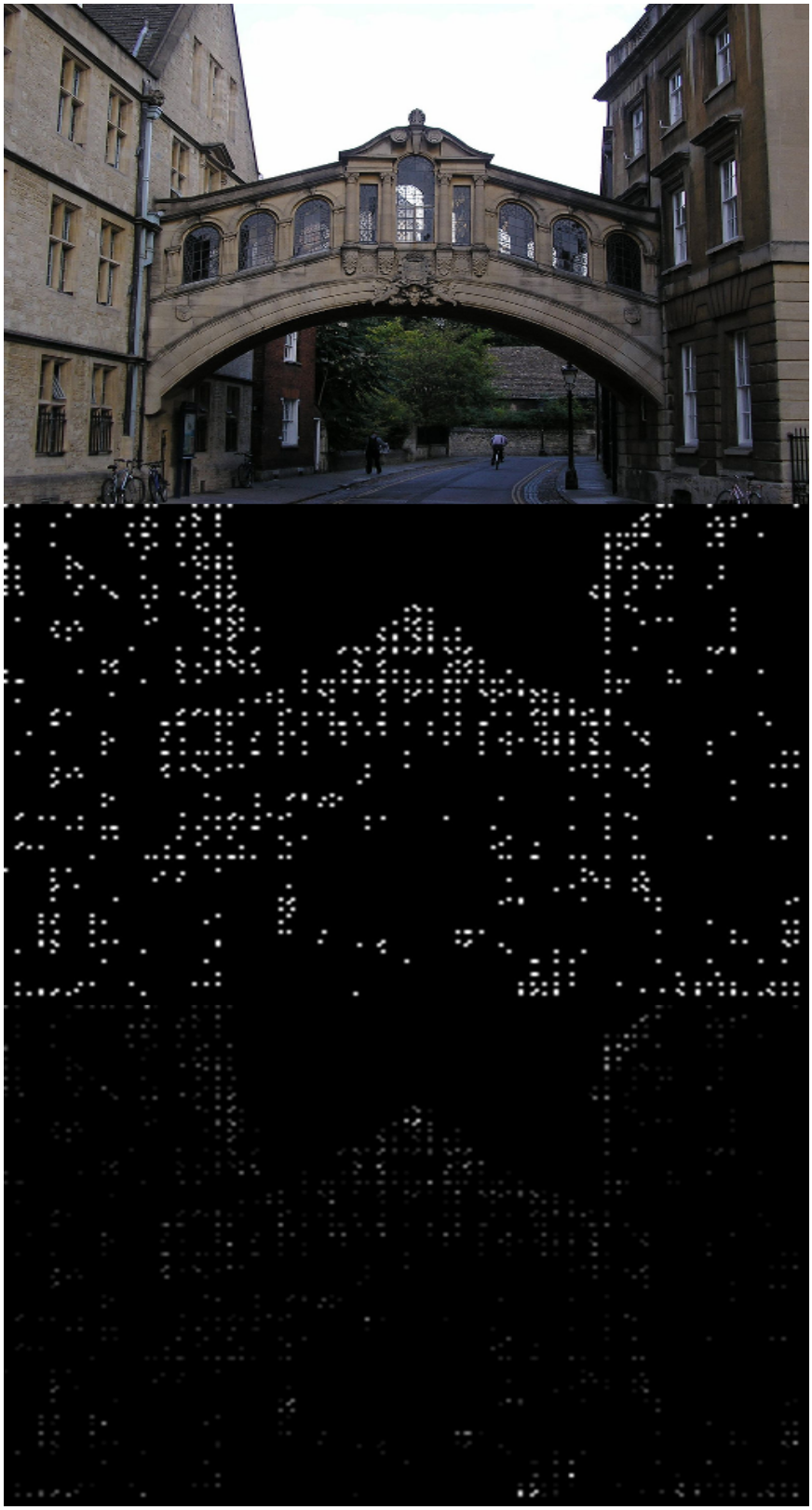
- DIR [7]: globalni deskriptor koji postiže najsuvremenije performanse u nekoliko postojećih skupova podataka. DIR deskriptori su 2048-dimenzionalne značajke ekstrahirane nad kovolucijskom mrežom ResNet101 [4],
- siaMAC [20]: globalni deskriptor koji postiže visoke performanse nad postojećim skupovima podataka. Temelji se na 512-dimenzionalnim značajkama ekstrahiranih iz VGG16 [21] mreže,
- LIFT [10]: nedavno predloženi cjevovod (engl. *pipeline*) za pronalaženje značajki gdje se odabir ključnih točaka, orijentacija i opisivanje uče zajedno. Značajke su 128-dimenzionalne.

Metode	Oxford5k - mAP(%)
ResNet50 (osnovica)	10.51%
ResNet50 + FT	35.38%
LIFT*	54.00%
siaMAC*	77.10%
DIR	86.10%
Resnet50 + FT + ATT (DELF - autori)	90.00%
Resnet50 + FT + ATT (DELF - ovaj rad)	90.84%

Tablica 2. Usporedba performansi metoda pri evaluaciji nad Oxford5k skupom podataka u mAP(%). FT predstavlja fino podešavanje mreže ResNet50, a ATT predstavlja model pažnje. Predzadnji redak predstavlja performanse modela autora rada, dok posljednji redak predstavlja performanse modela implementiranog u ovome radu. Metode označene zvjezdicom (*) trenirane su na drugačijem skupu podataka u odnosu na skup podataka korišten u ovome radu.

U Tablica 2. može se vidjeti usporedba različitih metoda. Kao osnovica (engl. *baseline*) korištena je mreža ResNet50 trenirana na ImageNet-u. FT predstavlja fino podešavanje mreže ResNet50, a ATT predstavlja model pažnje. Autori rada [5] za evaluaciju dodatno su implementirali sustav za pretraživanje slikovnih baza prema slikovnom sadržaju koji sažima značajke različitim metodama što dovodi do gubitka informacija. Performanse metoda LIFT, siaMAC i DIR uzete su iz originalnih radova ili iz [5]. Predzadnji redak Tablica 2. predstavlja performanse modela autora rada, dok posljednji redak predstavlja performanse modela implementiranog u ovome radu. Metode označene zvjezdicom (*) trenirane su na drugačijem skupu podataka u odnosu na skup podataka korišten u ovome radu.

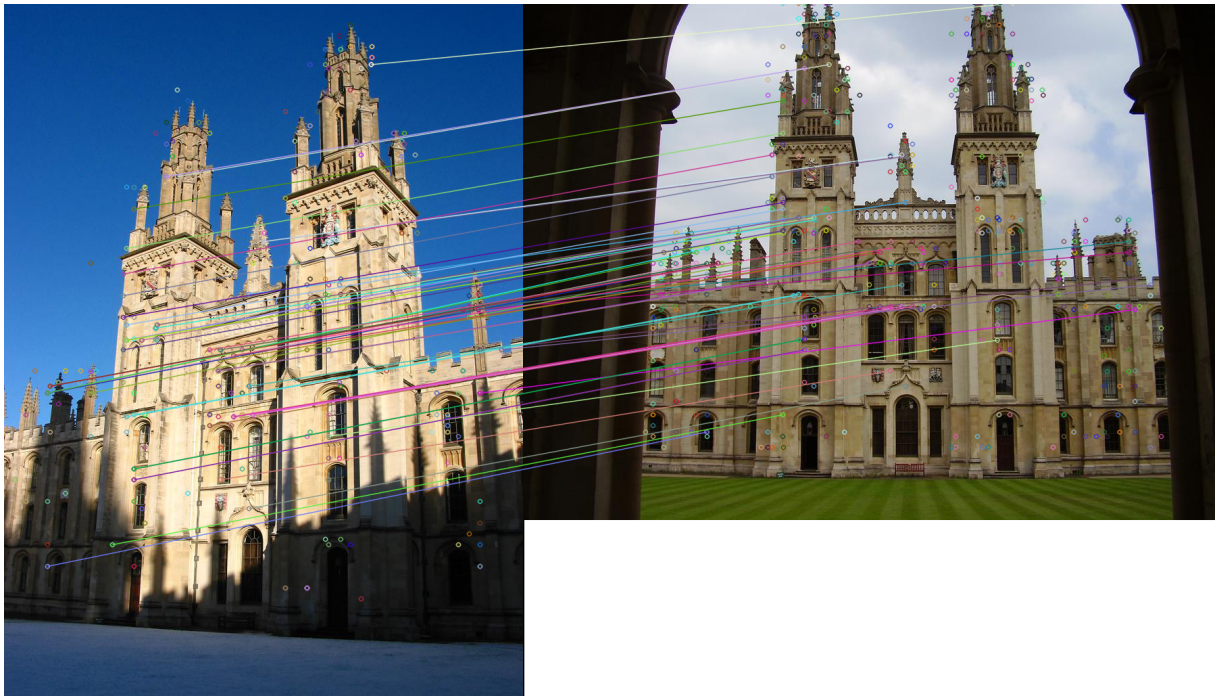
Na slikama 5.8 i 5.9 nalazi se vizualizacija 1000 značajki koje su na izlazu modela pažnje imale najveću vrijednost. Radi bolje vizualizacije, nad slikama s označenim lokaliziranim značajkama provodi se sažimanje maksimalnom vrijednošću s velikim kernelom (5×5 ili više) i korakom veličine kernela te se onda slika skalira na dvostruko veću. Na slici 5.9 može se vidjeti kako model pažnje odabire značajke nerelevantnih informacija sa slike (dijete, ljudi...), ali im uspješno pridružuje manje vrijednosti pažnje u odnosu na one relevantnije (toranj, okolna arhitektura...). Na srednjoj slici, 1000 značajki sa najvećom vrijednošću pažnje označeno je bijelim pikselom, a na donjoj slici vrijednosti tih istih bijelih piksela skalirane su s obzirom na korespondentnu vrijednost pažnje.



Slika 5.8: Gore - slika upita, Sredina - 1000 značajki sa najvećom vrijednošću pažnje označenih bijelim pikselom, Dolje - 1000 značajki sa najvećom vrijednošću pažnje čija je vrijednost bijelog piksela skalirana s obzirom na vrijednost pažnje.

Slika 5.9: Gore - slika upita, Sredina - 1000 značajki sa najvećom vrijednošću pažnje označenih bijelim pikselom , Dolje - 1000 značajki sa najvećom vrijednošću pažnje čija je vrijednost bijelog piksela skalirana s obzirom na vrijednost pažnje.

Slijede slike 5.10, 5.11, i 5.12 gdje se može vidjeti uparivanje točaka jedne slike upita u odnosu na jednu slučajnu sliku iz skupova *good*, *OK* i *junk*. Nakon što su odabrane ključne točke za pojedinu sliku, koristeći proizvoljnu metodu najbližih susjeda (kDTree [22] u sklopu ovoga rada) uparaju se točke čije su značajke međusobno blizu. Točke nakon toga prolaze i geometrijsku verifikaciju koristeći algoritam RANSAC. Parovi točaka (jedna točka sa slike upita, jedna sa slučajne slike iz pojedinih skupova) koji "prežive" geometrijsku verifikaciji su ista točka/značajka. Na slikama su povučene linije između korespondentnih istih točaka. Što je broj zajedičkih točaka veći, sustav je sigurniji da slike pripadaju istoj klasi. Upravo je broj zajedničkih točaka mjera sličnosti slika.



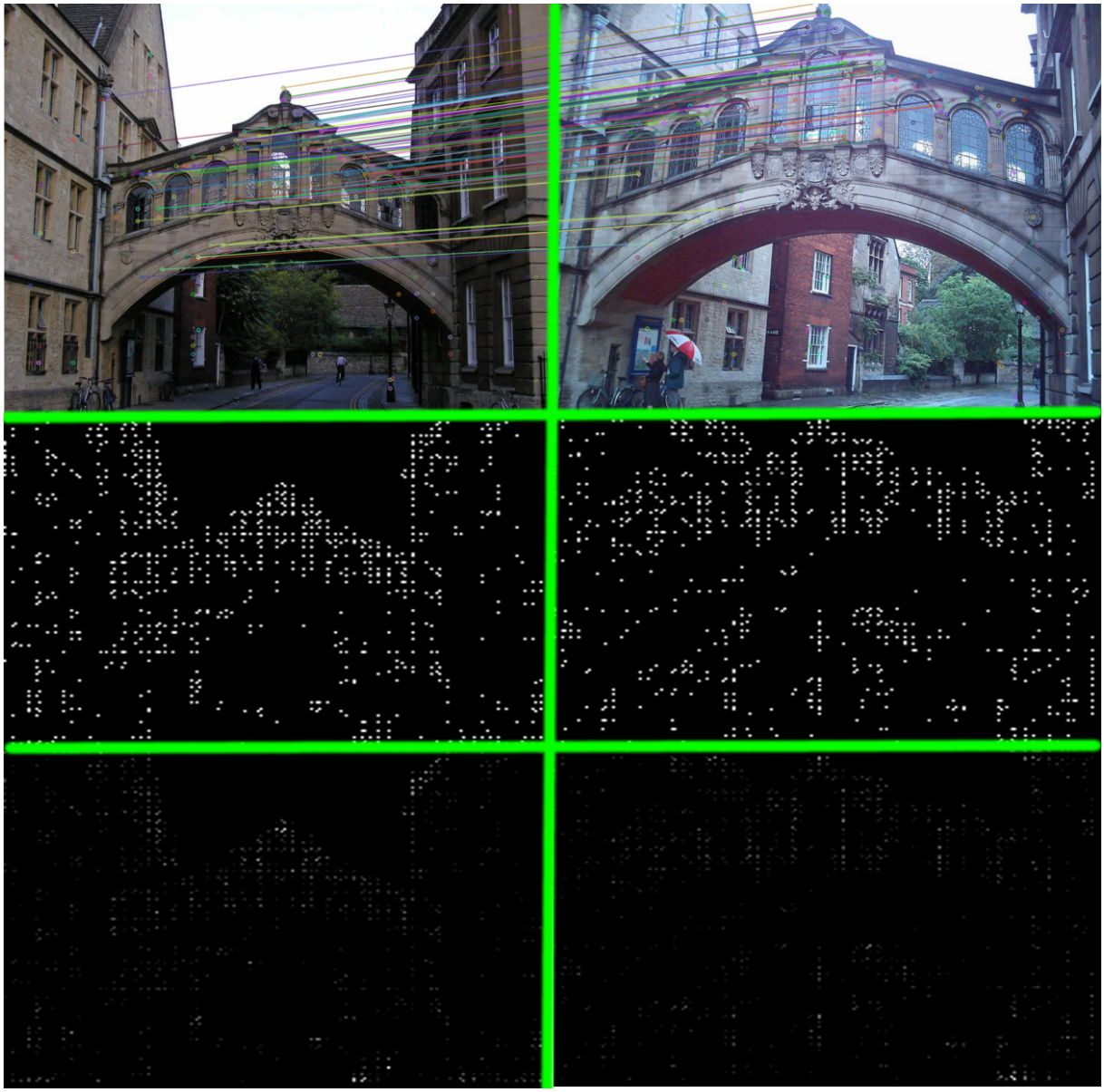
Slika 5.10: Podudaranje ključnih točaka za sliku upita i slučajnu sliku iz odgovarajućeg skupa good

Slika 5.11: Podudaranje ključnih točaka za sliku upita i slučajnu sliku iz odgovarajućeg skupa OK

Slika 5.12: Podudaranje ključnih točaka za sliku upita i slučajnu sliku iz odgovarajućeg skupa junk

Kao što je i očekivano, broj podudaranja smanjuje se promjenom "kvalitete" slike (*good, ok, junk*). Model i dalje uspješno pronalazi podudaranja u ključnim točkama unatoč promjeni perspektive, skale i/ili rotacije. Može se vidjeti u potpunosti razumljivo krivo podudaranje točaka dva prozora na slici 5.12. Zbog relativno velikih receptivnih polja, centri istih (lokalizirane značajke) mogu biti locirani izvan znamenitosti.

U nastavku se nalaze primjeri slika podudaranja ključnih točaka te vizualizacija korespondirajućih lokaliziranih značajki.



Slika 5.13: Vizualizacija podudaranja ključnih točaka te korespondirajućih lokaliziranih značajki. Lijevi stupac predstavlja sliku upita, a desni odgovarajuću sliku sa najviše zajedničkih točaka. Od gore prema dolje: slika upita, 1000 lokaliziranih značajki sa najvećom vrijednošću pažnje označenih bijelim pikselom, 1000 lokaliziranih značajki sa najvećom vrijednošću pažnje označenih bijelim pikselom čija je vrijednost skalirana s obzirom na vrijednost pažnje.

Slika 5.14: Vizualizacija podudaranja ključnih točaka te korespondirajućih lokaliziranih značajki. Lijevi stupac predstavlja sliku upita, a desni odgovarajuću sliku sa najviše zajedničkih točaka.

6. Zaključak

Pretraživanje slikovnih baza prema slikovnom sadržaju je važan zadatak računalnog vida. U posljednje vrijeme, veliki uspjeh u tom području ostvaruju pristupi temeljeni na naučenim značajkama do kojih dolazimo dubokim konvolucijskim modelima. U ovom radu opisane su duboke konvolucijske neuronske mreže, njihove gradivne jedinice kao i postupci vezani uz njihovu izgradnju. Fokus rada je model DELF (DEep Local Features), novi deskriptor lokalnih značajki dizajniran specifično za pretraživanje slikovnih baza prema slikovnom sadržaju [5]. DELF je učen slabim nadzorom koristeći samo oznake klasa na razini slike i duboko je povezan s mehanizmom modela pažnje za odabir semantički bitnih značajki. U predloženom potpunom konvolucijskom modelu dovoljan je jedan unaprijedni prolaz slike kroz mrežu za dobivanje ključnih točaka i deskriptora. Model je implementiran u razvojnom okviru Tensorflow s podrškom za CUDA paralelno programiranje. Za pravilnu evaluaciju uveden je skup podataka znamenitosti Oxford5k. Slike iz skupa podataka Oxford5k nisu korištene pri treniranju te je organiziran na način da ga je povoljno koristiti za evaluaciju sustava za pretraživanje slikovnih baza prema slikovnom sadržaju. Evaluacijom se pokazuje da DELF ostvaruje izvanredne performanse.

U budućem radu bilo bi zanimljivo iskoristiti izlaz nekog drugog konvolucijskog sloja prethodno opisane potpuno konvolucijske neuronske mreže ResNet50 (primjerice izlaz sloja *conv5_x* prikazan u Tablica 1). Bilo bi interesantno isprobati i neke druge mreže i arhitekture te možda čak isprobati drugačiju arhitekturu modela pažnje (dodavanje dodatnih konvolucijskih slojeva i sl.) te usporediti performanse sa DELF-om.

7. Literatura

[1] Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning, 2016. http://www.deeplearningbook.org.

[2] Marko Čupić. Optimizacija parametara modela, 2018. www.zemris.fer.hr/~ssegvic/du/du3optimization.pdf.

[3] C. Szegedy, S. Ioffe. Batch normalization: Accelerating deep network training byreducing internal covariate shift, 2015. https://arxiv.org/pdf/1502.03167v3.pdf.

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

[5] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, Bohyung Han. Large-Scale Image Retrieval with Attentive Deep Local Features, 2016. https://arxiv.org/abs/1612.06321.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database, 2009. http://www.image-net.org/papers/imagenet_cvpr09.pdf.

[7] Artem Babenko, Anton Slesarev, Alexandr Chigorin, Victor Lempitsky. Neural Codes for Image Retrieval, 2014. https://arxiv.org/abs/1404.1777.

[8] Charles Dugas, Yoshua Bengio, Francois Belisle, Claude Nadeau, Rene Garcia . Incorporating Second-Order FunctionalKnowledge for Better Option Pricing, 2002. http://www.iro.umontreal.ca/~lisa/publications2/index.php/attachments/single/83.

[9] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints, 2004. https://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/lowe_ijcv2004.pdf.

[10] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, Pascal Fua. LIFT: Learned Invariant Feature Transform, 2016. https://arxiv.org/abs/1603.09114.

[11] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen,Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, SanjayGhemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard,Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg,Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, MikeSchuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, PaulTucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals,Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, Xiaoqiang Zheng.. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. https://www.tensorflow.org.

[12] http://www.numpy.org

[13] https://matplotlib.org

[14] https://opencv.org

[15] Sergio Guadarrama, Nathan Silberman. "TensorFlow-Slim: a lightweight library for defining, training and evaluating complex models in TensorFlow", 2016. https://github.com/tensorflow/tensorflow/tree/master/tensorflow/contrib/slim.

[16] George A. Miller, Richard Beckwith, Christiane Fellbaum,Derek Gross, Katherine Miller. Introduction to WordNet: An On-line Lexical Database, 1993. http://wordnetcode.princeton.edu/5papers.pdf.

[17] Albert Gordo, Jon Almazan, Jerome Revaud, Diane Larlus. Deep Image Retrieval: Learning global representations for image search, 2016. https://arxiv.org/abs/1604.01325.

[18] Martin A. Fischler, Robert C. Bolles . Random Sample Consensus: A Paradigm for Model Fitting with Apphcatlons to Image Analysis and Automated Cartography, 1981. https://www.sri.com/sites/default/files/publications/ransac-publication.pdf.

[19] Martin A. Fischler, Robert C. Bolles . Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, 1981. https://www.sri.com/sites/default/files/publications/ransac-publication.pdf.

[20] Filip Radenović, Giorgos Tolias, Ondřej Chum. CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples, 2016. https://arxiv.org/abs/1604.02426.

[21] Karen Simonyan, Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014. https://arxiv.org/abs/1409.1556.

[22] Jon Louis Bentley. Multidimensional Binary Search Trees Used for Associative Searching , 1975. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.160.335&rep=rep1&type=pdf.

Konvolucijske reprezentacije za dohvat slika na temelju sadržaja

Sažetak

Pretraživanje slikovnih baza prema slikovnom sadržaju je važan zadatak računalnog vida. U posljednje vrijeme, veliki uspjeh u tom području ostvaruju pristupi temeljeni na naučenim značajkama do kojih dolazimo dubokim konvolucijskim modelima. U radu su opisane duboke neuronske mreže s fokusom na konvolucijske neuronske mreže. Prikazan je i implementiran model DELF u razvojnom okviru Tensorflow. Model je treniran na skupu podataka znamenitosti i evaluiran je na podatkovnom skupu Oxford5k. Na kraju su prikazani dobiveni rezultati s opisima i slikama.

Ključne riječi: dohvat slika na temelju sadržaja, duboke konvolucijske neuronske mreže, prijenos znanja, model pažnje.

Convolutional representations for content-based image retrieval

Abstract

Content-based image retrieval is an important task in the field of computer vision. Lately, great success in this area is acomplished using deep convolutional features. In this thesis, deep neural networks are described with emphasis on convolutional networks. The DELF model is introduced and implemented using the Tensorflow library. The model was trained on a landmark dataset and evaluated on the Oxford5k landmarks dataset. Finally, results with descriptions and images are presented.

Keywords: content-based image retrieval, deep convolutional networks, transfer learning, attention model.