

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

**Ekstrahiranje značajki slika pomoću
dubokog učenja u svrhu poboljšanja
sustava preporuke slika**

Toni Vlaić, Viran Ribić

Zagreb, travanj 2018.

Ovaj rad izrađen je u Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu pod vodstvom doc. dr. sc. Marina Šilića i predan je na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2017./2018.

SADRŽAJ

1. Uvod	1
1.1. Opći i specifični ciljevi rada	2
1.2. Programska i hardverska podrška	3
1.2.1. Programska podrška	3
1.2.2. Hardverska podrška	4
2. Modeli za sažimanje	5
2.1. Konvolucijske mreže	5
2.1.1. Konvolucijski slojevi	6
2.1.2. Slojevi sažimanja	8
2.1.3. Potpuno povezani slojevi	9
2.2. Analiza glavnih komponenti	10
2.3. VGG	12
2.4. Inception v4	12
3. Rezultati i rasprava	15
3.1. Metrike sličnosti	15
3.2. Predtrenirana arhitektura VGG	16
3.3. Predtrenirana arhitektura Inceptionv4	18
3.4. T-SNE prikaz smanjenog skupa podataka	23
4. Demo preporučitelj po sadržaju	25
4.1. Tijek rada	25
4.2. Implementacijski detalji	30
5. Zaključak	31
Literatura	34

POPIS TABLICA

3.1. Primjer uštede prostora	23
--	----

POPIS SLIKA

2.1.	Primjer ulazne slike 32x32x3 i rezultata izlaza nakon jednog konvolucijskog sloja dubine 5 [1]	6
2.2.	Primjer računanja izlaza (Lijevo) iz ulaza koji sadrži jedan kanal (Desno) pomicanjem prozora po kanalu [2]	7
2.3.	Utjecaj iznosa pomaka filtera na izlaze s prikazanim težinama na desnoj strani slike [1]	7
2.4.	Povećanje receptivnog polja s dubinom mreže [3]	8
2.5.	Sloj sažimanja s receptivnim poljem veličine 2x2 i pomakom 2 [1]	9
2.6.	Primjer sažimanja maksimalnom vrijednošću filterom 2x2 i pomakom 2 [1]	9
2.7.	Potpuno povezana unaprijedna neuronska mreža	10
2.8.	Određivanje glavnih komponenti podatkovnog skupa	10
2.9.	Transformirani podatkovni skup	11
2.10.	Podatkovni skup i crveno odabrane glavne komponente [4]	11
2.11.	Konfiguracije VGG tima [5]. Parametri konvolucije su prikazani kao conv<dimenzije filtera>-<broj kanala>, a nakon svake se nalazi ReLU aktivacija koja nije prikazana	12
2.12.	Inception v4 [6]	13
2.13.	Inception-A blok [6]	14
2.14.	Reduction-A blok (za Inception v4 k=192, l=224, m=256, n=384) [6] . . .	14
3.1.	Primjer euklidske udaljenosti [7]	15
3.2.	Primjer kosinusne udaljenosti [7]	16
3.3.	Top 12 sličnih rezultata za sliku na vrhu	17
3.4.	Slika na temelju koje se rade preporuka (lijevo) i slika u top 5 koja nije ispravna (desno)	18
3.5.	Top 12 sličnih rezultata za sliku na vrhu	19
3.6.	Top 12 sličnih rezultata za sliku na vrhu	20
3.7.	Top 12 sličnih rezultata za sliku na vrhu	21
3.8.	Top 12 sličnih rezultata za sliku na vrhu	22
3.9.	STL-10 dataset reprezentativni vektori prikazani u dvije dimenzije	24

4.1.	Stranica za registraciju novog korisnika	26
4.2.	Stranica za unos korisničkih podataka	26
4.3.	Galerija svih slika	27
4.4.	Galerija odabranih korisničkih slika	28
4.5.	Galerija preporučenih slika	28
4.6.	Primjer preporuka za sliku ptice	29
4.7.	Primjer preporuka za sliku purana	29
4.8.	Primjer preporuka za kombinaciju slika ptice i purana	30

1. Uvod

Digitalne slike čine jedinstveni format kojim se može opisati veliki broj različitih informacija o stvarnom svijetu. Sadržaj slike može uključivati tekstualne dokumente, satelitske snimke, medicinske zapise, ljudske interakcije, promet na cestama ili u zraku. Navedene su samo neke od kategorija od interesa koje možemo zabilježiti u digitalnom formatu, zajednički za sve interpretacije. Trodimenzionalni matrični zapis u računalu može opisati ogroman broj pojava koje susrećemo u stvarnom svijetu. Sve informacije koje čovjek razlučuje kroz vid mogu biti predstavljene u računalu u obliku informacije o intenzitetu svjetlosti i odnosu među susjednim pikselima. Reprodukcija ljudskog vizualnog sustava nije trivijalna, ali pruža alternativni pogled na računalne sustave inspirirane biološkim sustavima. Dva nova trenda u računarstvu su posljednjih godina potaknula velik broj istraživanja na području analize slike: enormne količine podataka koje se generiraju u svijetu i napredak u oblikovanju inteligentnih sustava i umjetne inteligencije.

Danas slike čine veliki udio podataka na internetu. Dok je egzaktna informacija o postotku slika na internetu intraktabilna, provedene analize iz 2016. godine [8] predviđale su porast od trilijun slika u narednoj godini. Rast slikovnih podataka povezan je s brojem korisnika pametnih telefona u svijetu kao i brojem online usluga koje uključuju obradu i razmjenu fotografija. Rastući trendovi korisnika [9] i napreci u telekomunikacijskim tehnologijama omogućuju lakšu i bržu distribuciju medijskih datoteka nego ikad prije.

Volumen i protok podataka u današnjim sustavima učinio je bilo kakvu ručnu obradu slika neadekvatnom što je potaknulo istraživače na razmatranje novih metoda pri analizi velikih količina podataka. Da bi se podacima moglo upravljati potrebno je prvo razviti efikasne metode pretrage. Novonastala situacija pronalazi sličnosti s ranim danima interneta dok još nisu formirane adekvatne metode pretraživanja stranica. Izmjena informacija bila je znatno teža zbog nedostatka prikladnih sučelja koja bi običnim korisnicima mogla pružiti jednostavno korištenje. Tek su se pojavom algoritama za indeksiranje otvorile sve mogućnosti izmjene podataka na internetu. Izgradnjom adekvatnih algoritama za pretraživanje i ekstrakciju znanja postići ćemo bolje razumijevanje podataka i omogućiti njihovu efikasniju upotrebu.

Ben Silbermann, osnivač i direktor društvene mreže Pinterest, izjavio je kako će buduć-

nost internet pretraživača biti upravo u efikasnoj interpretaciji slika. Za razliku od današnjih internetskih pretraživača koje se prvenstveno oslanjaju na ključne riječi unutar stranice, takvi bi pretraživači bili sposobni razumjeti semantiku slike i izvaditi ključne koncepte povezane s upitima korisnika [10].

Društvene mreže samo su jedan od primjera gdje će razumijevanje semantike slika imati ključnu ulogu u narednim godinama. Potražnja za ekstrakcijom znanja iz slika primjetna je i u marketinškim [11], medicinskim [12] [13], državnim [14], sigurnosnim [15] i forenzičkim [16] primjenama.

Napredak sklopolja vratio je u fokus modele neuronskih mreža pošto su nove arhitekture grafičkih kartica bile dovoljno efikasne za ponovno eksperimentiranje s umjetnim neuronskim mrežama. Rezultat novog vala testiranja bile su duble mreže, sposobne uhvatiti kompleksnije odnose među pikselima slika čime je označen početak dubokog učenja. Prikupljeni podaci o slikama sada postaju predmet proučavanja ovih pametnih algoritama koji pronalaze nove primjene u akademiji i industriji.

Ovaj rad proučava postojeće arhitekture i modele dubokog učenja i istražuje njihove primjene u oblikovanju sustava preporuke slika ekstrakcijom njihovih značajki. U nastavku uvodnog poglavlja rada izlažu se opći i specifični ciljevi te programski paketi korišteni u implementaciji. Drugo poglavlje obrađuje modele za sažimanje opisujući arhitekturu dubokih konvolucijskih mreža i predtreniranih modela *VGG* i *Inception v4*. Također, u poglavlje je uključen opis algoritma analize glavnih komponenti koji je korišten za efikasniju reprezentaciju značajki. Slijedi ih odjeljak s rezultatima i raspravom o ispitanim metrikama sličnosti, ponašanju predtreniranih arhitektura i metodama za inspekciju kvalitete modela na našem zadatku. Četvrto poglavlje pokriva demonstraciju rada sustava na primjeru *web* aplikacije. Opisani su tijek korištenja i informacije o implementaciji. U posljednjem, petom poglavlju, dan je osvrt na ishode rada.

1.1. Opći i specifični ciljevi rada

Hipoteza ovog istraživanja bila je da se korištenjem dubokog učenja može razviti sustav koji uspješno stvara nisko dimenzionalne reprezentacije slika proizvoljnih veličina kako bi se iste mogle efikasnije grupirati i pretraživati prema njihovim semantičkim značajkama. Takve reprezentacije moguće je iskoristiti u sustavima za preporučivanje koji se temelje na sadržaju ili kao dodatak sustavima preporuke temeljenim na kolaborativnom filtriranju kako bi se poboljšale njihove performanse.

Cilj ovog istraživanja bio je ostvariti takav duboki model, ispitati različite načine ekstrakcije značajki te ispitati metrike kojima se ekstrahirane značajke mogu uspoređivati. Također za kraj potrebno je i razviti prototip jednostavnog sustava za preporučivanje temeljen na

sadržaju koji koristi značajke ekstrahirane razvijenim modelom.

1.2. Programska i hardverska podrška

1.2.1. Programska podrška

Ovaj projekt implementiran je u dva programska jezika: *Python* i *Javascript*. *Python* je popularni programski jezik visoke razine apstrakcije. Zbog svoje jednostavnosti i velikog broja paketa otvorenog koda, često je prvi izbor za efikasno provođenje eksperimenata i brzo prototipiranje novih sustava. Paketi korišteni pri razvoju jezgre sustava za preporučivanje su: *Tensorflow*, *OpenCV*, *NMSLIB*, *t-SNE*. Za izradu demo prezentacije sustava, poslužitelj je implementiran u *Django* razvojnog okviru s popratnim paketima za efikasnu konfiguraciju. *Javascript* je programski jezik ključan za programiranje klijenata kroz koje će korisnici isprobavati demo sustav za preporučivanje slika. Radni okvir za pisanje klijenta je *Angular 5*, dok se više informacija o implementaciji možete pronaći u poglavlju 4.

Python paketi

Tensorflow [17] - radni okvir za strojno učenje. Razvijen i podržavan od *Google-a*, *Tensorflow* je alat za izgradnju modela strojnog učenja s mnogo dodataka, poput automatske diferencijacije pri popularnim metodama optimizacije i akcelerirano učenje na grafičkim karticama. Velika fleksibilnost i dobra podrška učinili su ovaj alat popularnim izborom u akademiji i industriji. Visoka modularnost i fleksibilnost ostvarene su oblikovanjem modela grafom operacija. *Tensorflow* je centralna komponenta našeg projekta pomoću koje se provodi ekstrakcija značajki slike.

OpenCV (engl. *Open Source Computer Vision Library*) [18] - radni okvir za obradu slika. *OpenCV* sadrži veliki skup algoritama za obradu i manipulaciju nad slikama. Ovaj se paket u našem projektu koristi za predobradu slika koje se kasnije prosljeđuju *Tensorflow* modelu.

NMSLIB (engl. *Non-Metric Space Library*)[19] - paket za efektivnu pretragu na temelju sličnosti. Paket je oblikovan bez ikakvih prepostavki o matematičkim svojstvima prostora pretraživanja što je postignuto implementiranjem funkcija za generičke prostore bez metrike. Ovaj sveobuhvatni sustav razmatrao je i dimenzionalnost prostora koji su iznimno veliki te podržava aproksimativne metode pri pretraživanju takvih prostora. U našem projektu ovaj se paket pokazao korisnim pri provođenju eksperimenata nad različitim metodama ekstrakcije.

t-SNE (engl. *t-distributed Stochastic Neighbor Embedding*) [20] - modul paketa *sklearn* za vizualizaciju visoko dimenzionalnih podataka. *sklearn* obuhvaća skup alata za razne zadatke potrebne pri izradi inteligentnih sustava. On sadrži algoritme i metrike za česte zadatke klasifikacije, regresije, klasteriranja kao i redukcije dimenzionalnosti, odabira modela

i analize podataka. Jedan od modula uključenih u ovaj set alata je i *t-SNE* modul koji služi za pretvorbu vektorskih podataka visokih dimenzija u zapis pogodan za iscrtavanje na grafu. Postupak odabira željenih dimenzija vrši se minimizacijom Kullback-Leiber divergencije između zajedničke vjerojatnosti sažetka i izvornog podatka.

Django [21]- radni okvir za izradu *web* poslužitelja. *Django* omogućava brzi razvoj projekta s nizom gotovih komponenti i mogućnošću proširivanja osnovnih funkcionalnosti kako bi se ostvarilo željeno ponašanje. Radni okvir uključuje autentifikaciju, posluživanje *web* stranica i *REST* (engl. *REpresentational State Transfer*) zahtjeva. *Django* se, u našem projektu, koristi isključivo za pohranu, obradu i posluživanje podataka te kao jedinstvena točka za pristup bazi podataka.

JavaScript paketi

Angular 5 [22] - radni okvir za izradu *web* klijenata. *Angular 5* oslanja se na *TypeScript* paket pri oblikovanju svih komponenti. To Angular čini strogo tipiziranim i omogućava pišanje izoliranih komponenti koje se mogu povezivati. Posljedica je jasna paradigma koja pruža sve funkcionalnosti potrebne za razvoj *web* stranice uz najveću moguću fleksibilnost pri oblikovanju izgleda i ponašanja komponenti stranice. U našem projektu Angular je korišten za dizajn *web* klijenta kojim korisnik može isprobati ponašanje sustava za različite korisničke odabire slika.

1.2.2. Hardverska podrška

Implementacija rješenja i ispitivanje napravljeno je na računalu s operativnim sustavom *Ubuntu 16.04*. Od važnijih komponenti koje utječu na brzinu i mogućnost reprodukcije rezultata bitno je naglasiti da računalo ima grafičku karticu *GTX Titan Pascal* s 12GB memorije, *Intelov i7* procesor sa šest jezgri i 32GB radne memorije što je omogućilo obradu velikog podatkovnog skupa u razumnom vremenu.

2. Modeli za sažimanje

2.1. Konvolucijske mreže

Konvolucijske neuronske mreže su postale popularne kao sredstvo za obradu slika nakon pobjede Alexa Krizhevsky-og na natjecanju 2012 *ILSVRC* (engl. *ImageNet Large-Scale Visual Recognition Challenge* [23]) sa svojom konvolucijskom neuronskom mrežom danas poznatijom pod imenom *AlexNet*. Od tada je njihova popularnost kao sredstvo za obradu slika naglo porasla te su danas konvolucijske arhitekture najdominantnije u području obrade slika.

Računalo promatra sliku kao trodimenzionalnu matricu dimenzija $X*Y*Z$ gdje su X i Y visina i širina slike, a dimenzija Z predstavlja broj kanala promatrane slike (uobičajeno crveni-zeleni-plavi kanal, poznatije kao RGB). Uzimajući u obzir da su ulazni podaci modela konvolucijske mreže uobičajeno slike, odnosno trodimenzionalne matrice, ona svoje neurone kojima obrađuje primljene podatke također uređuje u tri dimenzije, širina, visina i dubina koja se može smatrati i brojem kanala kao u slikama, no bez limitacija na broj istih.

Jedna duboka konvolucijska mreža se uobičajeno gradi od konvolucijskih slojeva, slojeva sažimanja i potpuno povezanih slojeva unaprijedne neuronske mreže. Između svih slojeva u mreži nalazi se tzv. aktivacijska funkcija, čiji je cilj uvesti nelinearnost u model.

Aktivacijske funkcije su hiperparametri svake mreže i bitne su zbog unošenja nelinearnosti. U slučaju nedostatka aktivacija mreža od više slojeva reducirala bi se na afinu transformaciju koja preslikava ulazne vrijednosti u izlazne. Aktivacijska funkcija završnog sloja ovisi o tipu zadatka kojeg se mreža uči rješavati. Mreža trenirana za aproksimaciju funkcije neće raditi transformacije nad izlazom, već će aktivacijska funkcija izlaza biti funkcija identiteta (2.1). Drugi primjer bio bi zadatak klasifikacije za koji izlaz mora imati probabilističku interpretaciju, a ona se postiže sigmoidom (2.2) u slučaju dvije klase ili softmax funkcijom (2.3) za više klase.

$$Identitet(x) = x \quad (2.1)$$

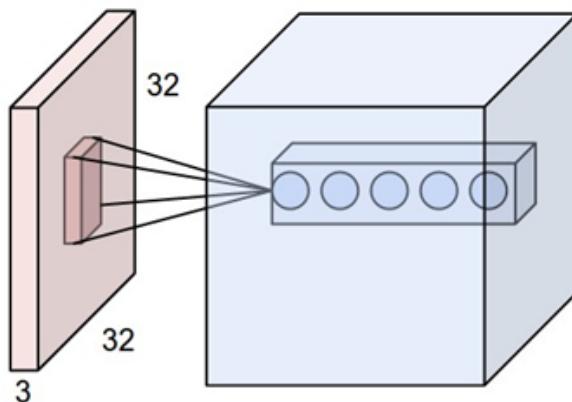
$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \quad (2.2)$$

$$jeSoftmax(x)_i = \frac{e^{x_i}}{\sum_j^N e^x} \quad \text{for } j = 1, \dots, N \quad (2.3)$$

2.1.1. Konvolucijski slojevi

Konvolucijski sloj odradjuje većinu posla u konvolucijskoj mreži. Novost u konvolucijskoj mreži je povezivanje svakog neurona s malom regijom prethodnog sloja umjesto potpunog povezivanja svih neurona između slojeva kao u potpuno povezanoj unaprijednoj mreži. Svaki konvolucijski sloj ima tri dimenzije i neuroni svakog kanala dijele filtere pomoću kojih računaju izlaze iz prethodnog sloja.

Uzmimo za primjer jedan konvolucijski sloj koji obrađuje CIFAR ([24]) skup podataka s dimenzijama slika $32 \times 32 \times 3$. Ulaz takvog sloja su slike $32 \times 32 \times 3$, a filteri mogu biti zadani kao $3 \times 3 \times 5$. Dimenzije 3×3 označavaju receptivno polje prozora koje pomiče po svakom kanalu ulaznog podatka, a broj pet označava novu dubinu nakon svih izračuna (Slika 2.1). Broj parametara takvog sloja iznosi broj izlaznih kanala (5) * površina prozora koji se pomiče (3×3) * broj ulaznih kanala (3), odnosno 135 i dodatnih 5 vrijednosti za pristranost po svakom izlaznom kanalu.

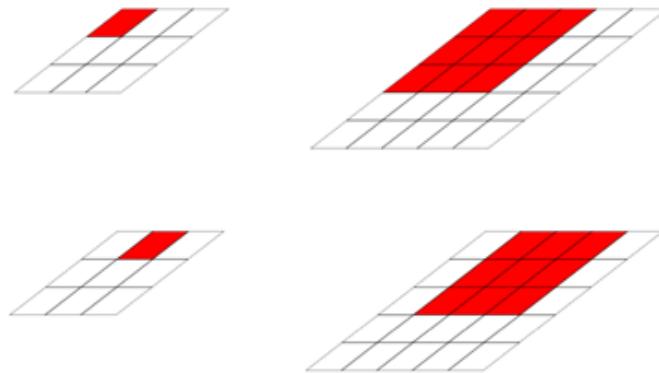


Slika 2.1: Primjer ulazne slike $32 \times 32 \times 3$ i rezultata izlaza nakon jednog konvolucijskog sloja dubine 5 [1]

Intuitivno konvolucijski slojevi će naučiti filtere koji će reagirati na određeni tip vizualnih svojstava, pa ima smisla dijeliti parametre filtera unutar svakog kanala neovisno o trenutno promatranom području slike.

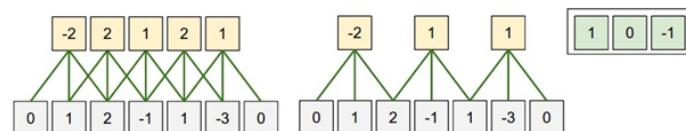
Svaki filter ima onoliko pomičnih prozora koliko prethodni sloj ima kanala, a svaki pomični prozor ima prethodno određenu površinu receptivnog polja, npr. 3 prozora površine 3×3 kao u gornjem primjeru koji se pomiču po svom zaduženom kanalu. Moguće je svakom konvolucijskom filteru definirati korak (engl. stride) kojim se prozor pomiče po kanalima prethodnog sloja. Jedan položaj filtera daje jednu vrijednost u jednom izlaznom kanalu. Iz-

laz filtera je suma svih izlaza pojedinačnih prozora po kanalu (Slika 2.2). Svaki prozor sa slike množi određenom težinom element na odgovarajućem mjestu te sumira dobivene vrijednosti svih prozora dobivajući vrijednost krajnjeg izlaza. Nakon izračuna svih vrijednosti sloja one se šalju u aktivacijsku funkciju i koriste se kao ulaz u sljedeće slojeve.



Slika 2.2: Primjer računanja izlaza (Lijevo) iz ulaza koji sadrži jedan kanal (Desno) pomicanjem prozora po kanalu [2]

Korak pomaka (engl. stride) definira za koliko elemenata će se pomicati filter, odnosno svaki prozor, pri računanju izlazne vrijednosti sljedećeg neurona. Veći korak utječe na dimenziju izlaznog sloja (Slika 2.3).

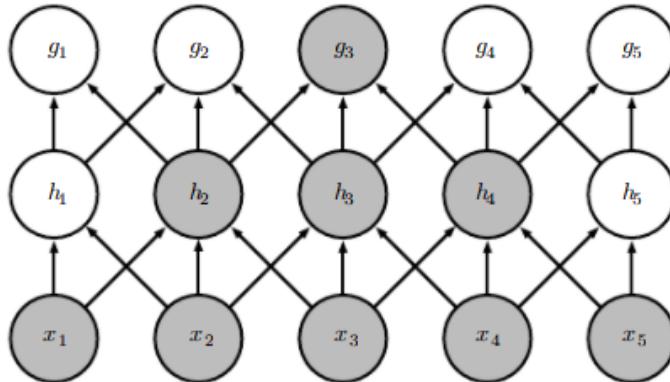


Slika 2.3: Utjecaj iznosa pomaka filtera na izlaze s prikazanim težinama na desnoj strani slike [1]

Osim koraka pomaka, na dimenziju izlaza utječe i površina receptivnog polja filtera. Veće receptivno polje uzrokovat će manji broj izlaza jer će se na ulaznu sliku moći posložiti manja količina prozora iz kojih se računaju izlazne vrijednosti. Uobičajena praksa je proširiti ulaznu sliku dodavanjem nula (engl. padding) na rubove kako bi se nakon konvolucije zadržala dimenzija izlaza jednaka ulazu. Količina nula koje je potrebno dodati ovisi o receptivnom polju i koraku konvolucije.

Dodavanjem slojeva receptivno polje novih neurona se povećava ovisno o receptivnom polju prošlih neurona. Primjerice, nakon dvije konvolucije s receptivnim poljem 3×3 i korakom jedan, neuroni u drugom sloju vide veću površinu originalne slike od neurona u sloju prije njih (Slika 2.4). Zbog toga se dublji slojevi konvolucijske mreže uče prepoznavati složenije vizualne značajke poput kotača, dok niži slojevi uče prepoznavati jednostavne značajke

poput rubova objekata.

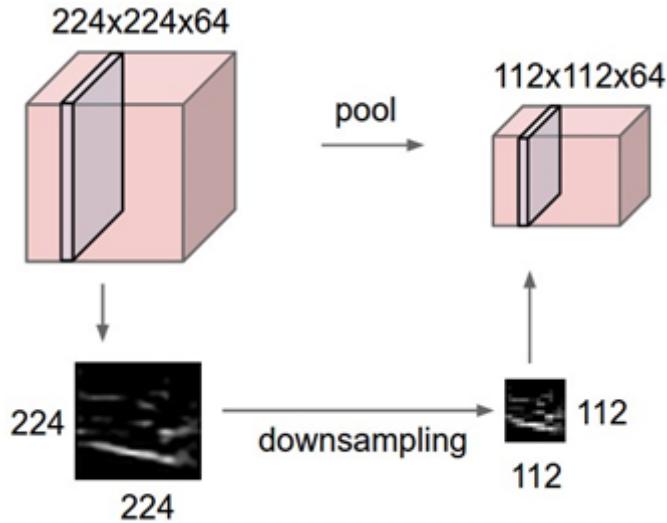


Slika 2.4: Povećanje receptivnog polja s dubinom mreže [3]

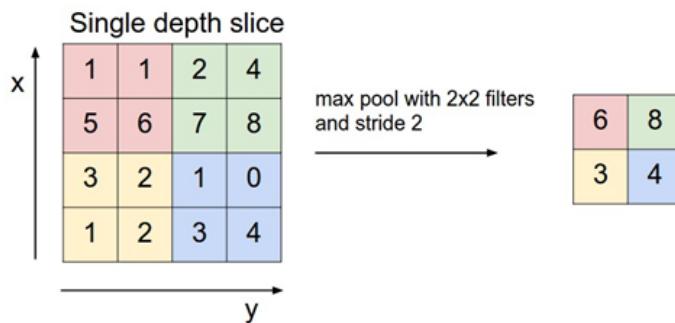
2.1.2. Slojevi sažimanja

Sloj sažimanja je vrlo jednostavan, ali i vrlo bitan dio konvolucijske mreže. Sažimanje obično računa neku statističku vrijednost nad receptivnim poljem poput maksimalne vrijednosti ili srednje vrijednosti. Svrha sloja sažimanja je smanjiti dimenzionalnost (Slika 2.5) i broj potrebnih parametara mreže. Dodatni pozitivan efekt smanjena broja parametara je i smanjenje vjerojatnosti da se takva mreža prenauči na ulazne podatke.

Osim reduciranja dimenzionalnosti, slojevi sažimanja unose invarijantnost na pomake u slici. Do intuitivnog objašnjenja prethodne tvrdnje možemo doći ako pretpostavimo da filter reagira na neki oblik u prethodnom sloju što prepoznajemo po većoj pozitivnoj vrijednosti izlaza od ostalih susjednih neurona (Slika 2.5). Budući da za klasifikaciju nije bitno znati gdje se oblik nalazi na slici nego nalazi li se na slici, može se iskoristiti sažimanje maksimalnom vrijednošću. Takvo sažimanje će u sljedeće slojeve propustiti samo maksimalni odziv u promatranom prozoru (Slika 2.6).



Slika 2.5: Sloj sažimanja s receptivnim poljem veličine 2×2 i pomakom 2 [1]

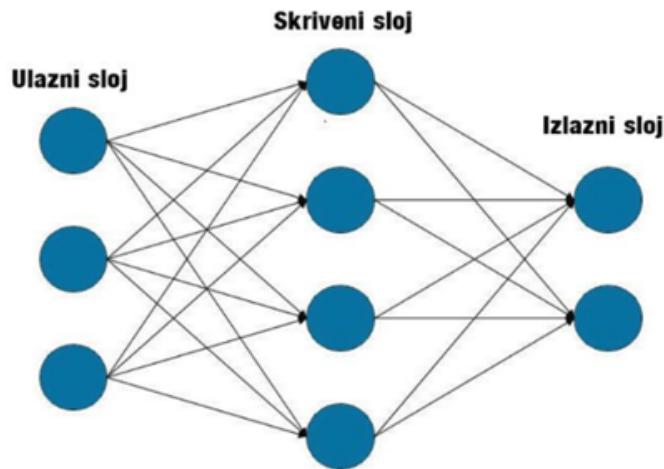


Slika 2.6: Primjer sažimanja maksimalnom vrijednošću filterom 2×2 i pomakom 2 [1]

2.1.3. Potpuno povezani slojevi

Potpuno povezani slojevi dolaze iz najjednostavnijeg oblika neuronskih mreža, potpuno povezanih unaprijednih mreža, zbog čega će ovi slojevi biti objašnjeni izravno objašnjavanjem načina rada te vrste arhitekture mreža.

Potpuno povezana unaprijedna neuronska mreža je građena od slojeva koji se razlikuju jedino po aktivacijskoj funkciji: ulazni sloj na koji stižu podaci čija je aktivacija funkcija identiteta, skrivenih slojeva na čijem se izlazu nalazi neka derivabilna nelinearna funkcija te izlaznog sloja s konačnim rezultatom izračuna s aktivacijom prikladnom za problem koji se nastoji riješiti (Slika 2.7). Svaki ulaz u neuron ima svoju težinu koja množi ulaz te se svi ulazi na kraju sumiraju i šalju u aktivacijsku funkciju. Rezultat aktivacijske funkcije zatim postaje ulaz u sljedeće slojeve ili predstavlja završni rezultat iz zadnjeg sloja.

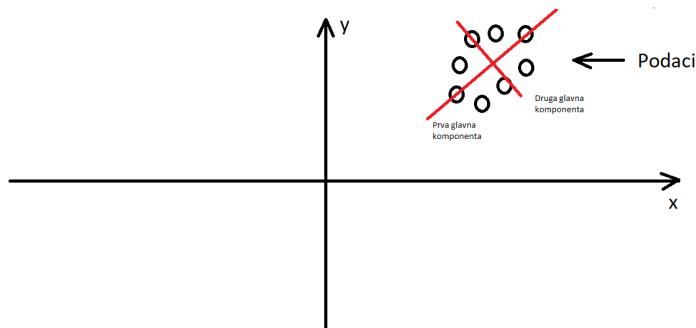


Slika 2.7: Potpuno povezana unaprijedna neuronska mreža

2.2. Analiza glavnih komponenti

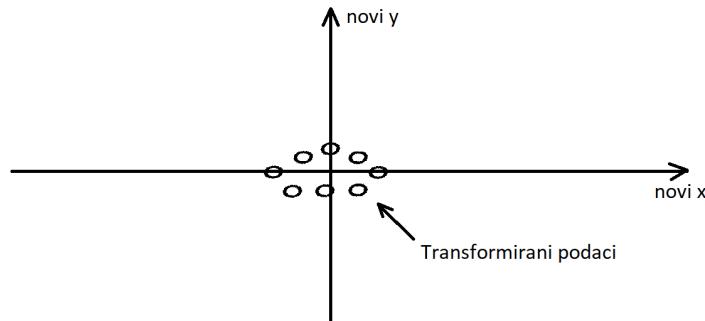
Analiza glavnih komponenti (engl. *Principal component analysis - PCA*) standardni je alat u modernoj analizi podataka - u različitim područjima od neuroznanosti do računalne grafike zbog toga što je jednostavna, neparametarska metoda za ekstrahiranje relevantnih informacija iz podatkovnog skupa [4].

Bez ulazeња u matematiku koja se može pronaći u radu [4], *PCA* se može objasniti laičkim terminima kao tehnika koja pronalazi temeljne varijable koje najbolje diferenciraju podatkovni skup (poznatije kao glavne komponente - engl. principal components). Glavne komponente skupa su dimenzije po kojima je naš podatkovni skup najviše raspršen (Slika 2.8).



Slika 2.8: Određivanje glavnih komponenti podatkovnog skupa

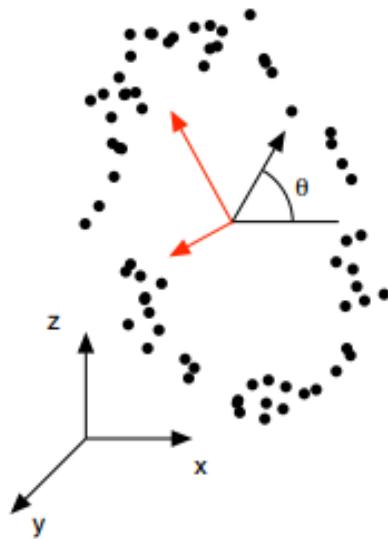
Nakon što je određen traženi broj glavnih komponenti, u slučaju sa slike njih dva, podatkovni skup se transformira u novi koordinatni sustav (Slika 2.9) čije su koordinatne osi dekorelirane.



Slika 2.9: Transformirani podatkovni skup

Iako se u gore navedenom primjeru dimenzionalnost podataka ne reducira (2D podaci ostaju u 2D prostoru), već se oni samo dekoreliraju, glavno područje primjene *PCA* tehnike je upravo redukcija dimenzionalnosti koja funkcioniра na skoro identičan način. Upotreba *PCA* metode za redukciju dimenzionalnosti razlikuje se po tome što se nakon određenih smjerova raspršenja oni poredaju silazno po svojstvenim vrijednostima te se odabere željena dimenzija na koju se podaci žele transformirati. Nakon što se završi transformacija na manji broj dimenzija ostale glavne komponente s nižim svojstvenim vrijednostima se gube.

Važno je naglasiti da je glavna motivacija iza *PCA* tehnike uklanjanje korelacije u podatkovnom skupu, odnosno uklanjanje zavisnosti drugog reda između podataka [3]. Način na koji *PCA* nastoji ostvariti taj cilj je pretpostavkom da su glavne komponente međusobno ortogonalne zbog čega *PCA* neće biti primjerena za podatkovne skupove poput onog na slici 2.10.



Slika 2.10: Podatkovni skup i crveno odabrane glavne komponente [4]

2.3. VGG

Arhitektura prvog isprobog modela datira iz 2014. godine, a dolazi sa sveučilišta Oxford čiji ju je tim koristio u *ImageNet* natjecanju. *VGG* tim (engl. *Visual Geometry Group*), po kojem je mreža dobila ime, ostvario je prvo mjesto iz zadatka lokalizacije objekata i drugo mjesto iz zadatka klasifikacije.

Njihov glavni doprinos je temeljita evaluacija arhitektura dubokih mreža koje koriste konvolucijske filtere malih dimenzija (3x3), čime su pokazali da se značajno poboljšanje u odnosu na tadašnje najbolje arhitekture može postići povećanjem dubine mreže na 16-19 slojeva dubine [5].

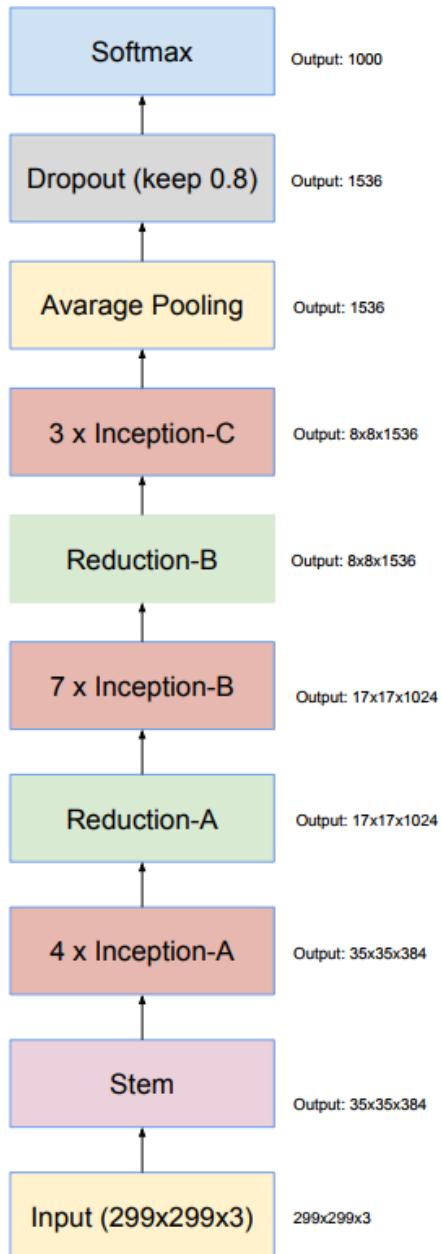
Nakon evaluiranja više konfiguracija mreža, *VGG* tim je pokazao da dublja konvolucijska mreža nadmašuje rezultate mreža s manje slojeva, zbog čega smo u našim ispitivanjima koristili njihovu najdublju mrežu u stupcu E sa slike 2.11.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Slika 2.11: Konfiguracije VGG tima [5]. Parametri konvolucije su prikazani kao conv<dimenzije filtera>-<broj kanala>, a nakon svake se nalazi ReLU aktivacija koja nije prikazana

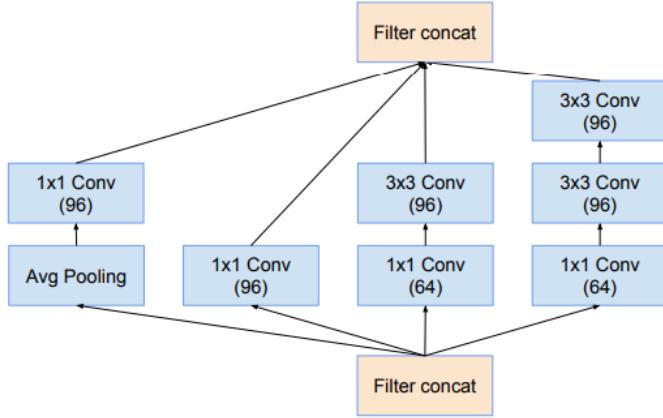
2.4. Inception v4

Inception v4 model je *Google*-ova četvrta iteracija *Inception* arhitekture koja se od *VGG* arhitekture najviše razlikuje po svojoj “širini”. Arhitektura mreže u skraćenom prikazu *Inception* blokova se može vidjeti na slici (Slika 2.12).



Slika 2.12: Inception v4 [6]

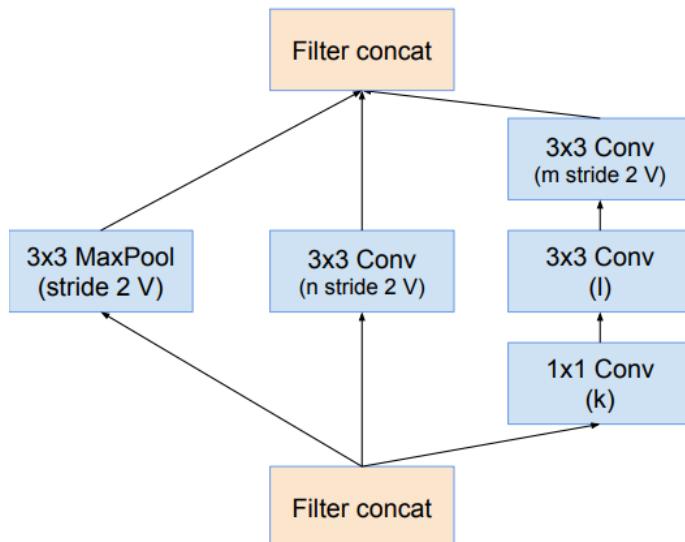
Inception blokovi su postali popularni nakon što je *Google*-ova tadašnja arhitektura *GoogLeNet* izgrađena od *Inception* blokova ostvarila dobre performanse na natjecanju ImageNet. Njihova glavna prednost je rješavanje problema odabira dimenzija filtera pri izgradnji mreže jer se sastoje od više grana konvolucija koje ulazni podatak obrađuju s različitim dimenzijama filtera. Zbog stabilaste razgranatosti, *Inception* blok iz ulaza može prepoznati komplikiranije uzorce jer svaka grana ulazni podatak promatra s različitim receptivnim poljem (Slika 2.13).



Slika 2.13: Inception-A blok [6]

Inception moduli, od kojih je građena mreža (A-C), obrađuju ulaze na sličan način. Zajednička karakteristika im je da izlazne dimenzije podatka ostaju jednake (npr. *Inception-A* na ulazu prima $35 \times 35 \times 384$ a na izlazu daje $35 \times 35 \times 384$), a razlikuju se po načinu obrade ulaznog podatka, točnije broja konvolucijskih slojeva u granama i njihovih parametara.

Osim *Inception* modula, mreža se sastoji i od *Reduction* blokova koji su, uz ekstrakciju značajki, zaslužni za smanjivanje dimenzionalnosti ulaznog podatka. Kao i *Inception* blokovi razlika između *Reduction-A* bloka i *Reduction-B* bloka je samo u parametrima grana i dizajnu grana, a oba bloka smanjuju širinu i visinu ulaznog podatka za faktor 2 korištenjem slojeva sažimanja ili konvolucija s korakom 2 (Slika 2.14).



Slika 2.14: Reduction-A blok (za Inception v4 $k=192$, $l=224$, $m=256$, $n=384$) [6]

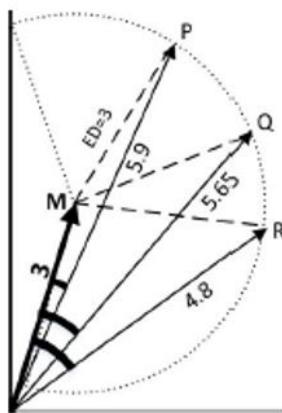
3. Rezultati i rasprava

3.1. Metrike sličnosti

Prilikom uspoređivanja sličnosti vektora isprobali smo dvije metrike, euklidsku i kosinusnu udaljenost.

Euklidska udaljenost definirana je kao L2 udaljenost između dva vektora (3.1), a raspon vrijednosti iznosi od nula, za točke koje se nalaze na istom mjestu u prostoru, do beskonačnosti. Negativna strana euklidske udaljenosti je zanemarivanje informacija o kutu između dvije točke. Primjer te situacije se može vidjeti na slici 3.1 gdje su točka P, Q i R jednako udaljene od točke M, no kada bismo uzimali informaciju o kutu u obzir najsličnija bi nam bila točka P [7].

$$EuclideanDistance(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3.1)$$

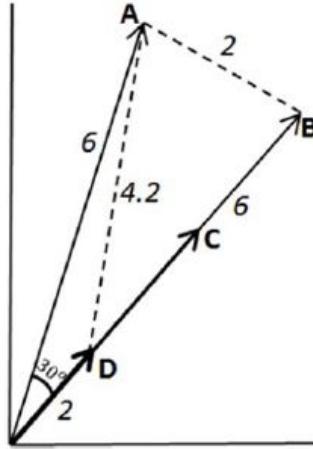


Slika 3.1: Primjer euklidske udaljenosti [7]

Kosinusna udaljenost je definirana kao jedan minus skalarni produkt dva vektora podijeljen s umnoškom njihovih normi (3.2), a raspon vrijednosti je od 2, za vektore koji gledaju u suprotnim smjerovima, do 0, za kolinearne vektore. Negativna strana kosinusne udaljenosti je zanemarivanje duljine vektora. Primjer se može vidjeti na 3.2 gdje su točke B, C i

D jednako udaljene od točke A jer zatvaraju jednak kut, no kada bi se promatrala njihova euklidska udaljenost dobili bismo različite rezultate [7].

$$\text{CosineDistance}(x, y) = 1 - \cos(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} \quad (3.2)$$



Slika 3.2: Primjer kosinusne udaljenosti [7]

Obje metrike su davale skoro identične rezultate uz male promjene poretku između vraćenih slika. Jedan od razloga zašto smo dobili takve rezultate je velika dimenzionalnost prostora kojeg vektori razapinju. Posljedica toga je da dvije točke koje se nalaze blizu u prostoru ujedno zatvaraju i maleni kut zbog čega se smatraju sličima u obje metrike.

Nakon ispitivanja rezultata metrika, odabrali smo kosinusnu udaljenost koja je davala zadovoljavajuće rezultate. Potencijalni problemi koji se mogu pojaviti kod te metrike u našem slučaju nisu pretjerano bitni budući da nam nije bitna duljina vektora već njegovo usmjerenje, odnosno relativan iznos aktivacija koje taj vektor opisuje. Primjerice, ako našem sustavu predamo sliku mačke i tražimo ostale slične slike, želimo vratiti sve vektore kojima su dominantni elementi koji detektiraju oči, njušku, uši mačke. Zbog toga nam je kosinusna udaljenost zadovoljavajuća jer vraća slike čiji reprezentativni vektori imaju jednak odnos dominantnih elemenata unutar vektora.

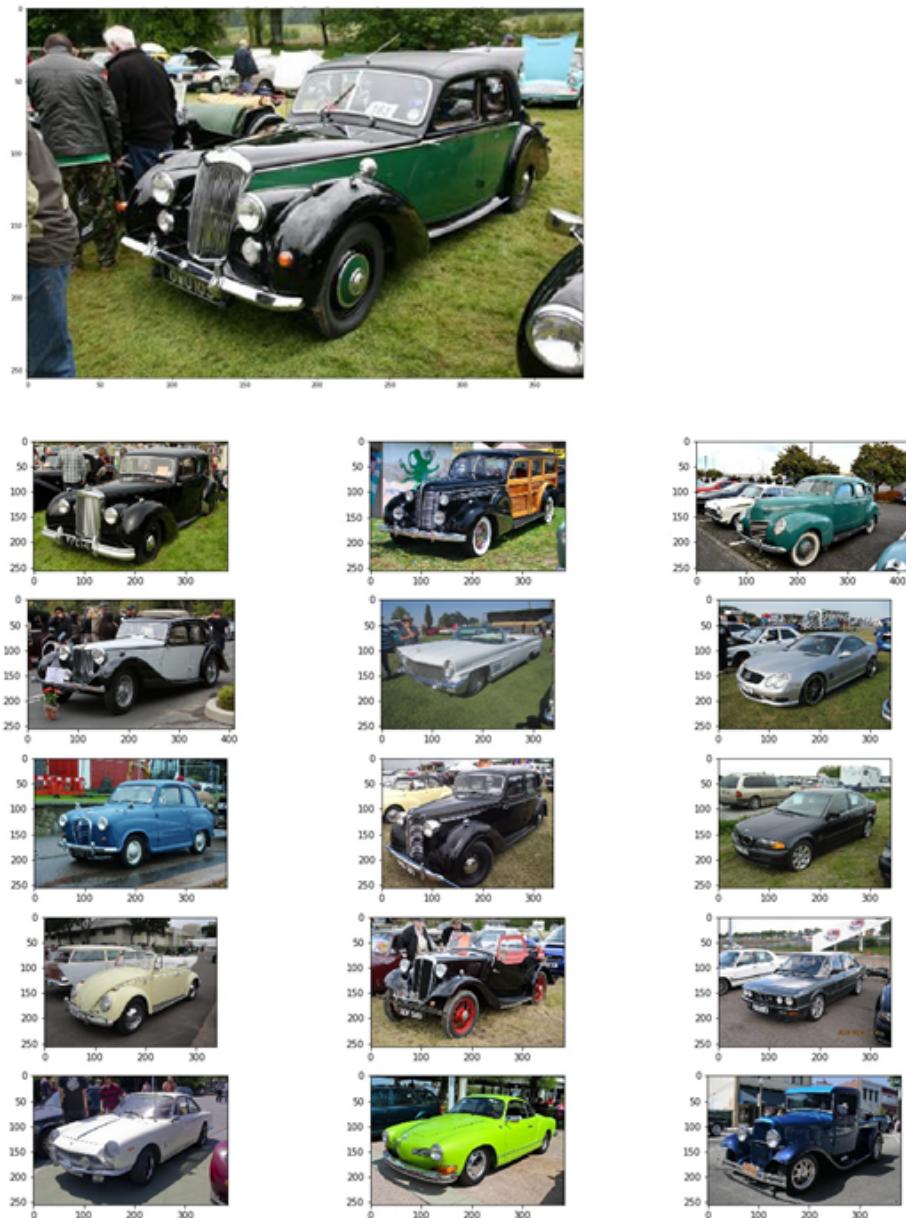
3.2. Predtrenirana arhitektura VGG

VGG arhitektura je prva isprobana mreža za ekstrakciju značajki, točnije njezin posljednji sloj konvolucijskog dijela mreže. Težinski koeficijenti mreže preuzeti su iz pobjedničke arhitekture ImageNet natjecanja čiji su filteri već naučeni prepoznavati komplikirane uzorke na ulaznim slikama.

Ulagana slika u mrežu je dimenzija 224x224x3, a finalni rezultat sloja kojeg koristimo je vektor od 25.000 elemenata. Budući da je krajnji cilj ostvariti reprezentaciju slike vektorom

manjih dimenzija, što bi omogućilo brže pretraživanje, rezultat konvolucijske mreže se *PCA* modelom dalje reducira na 3000 dimenzija. Za treniranje *PCA* modela nasumično je izabранo 50.000 slika kako bismo dobili što bolju procjenu glavnih komponenti.

Rezultati dobiveni na ovaj način za pojedine slike daju očekivane rezultate (Slika 3.3), dok za druge već u top 5 najsličnijih slika vidimo nepovezanost rezultata (Slika 3.4).



Slika 3.3: Top 12 sličnih rezultata za sliku na vrhu



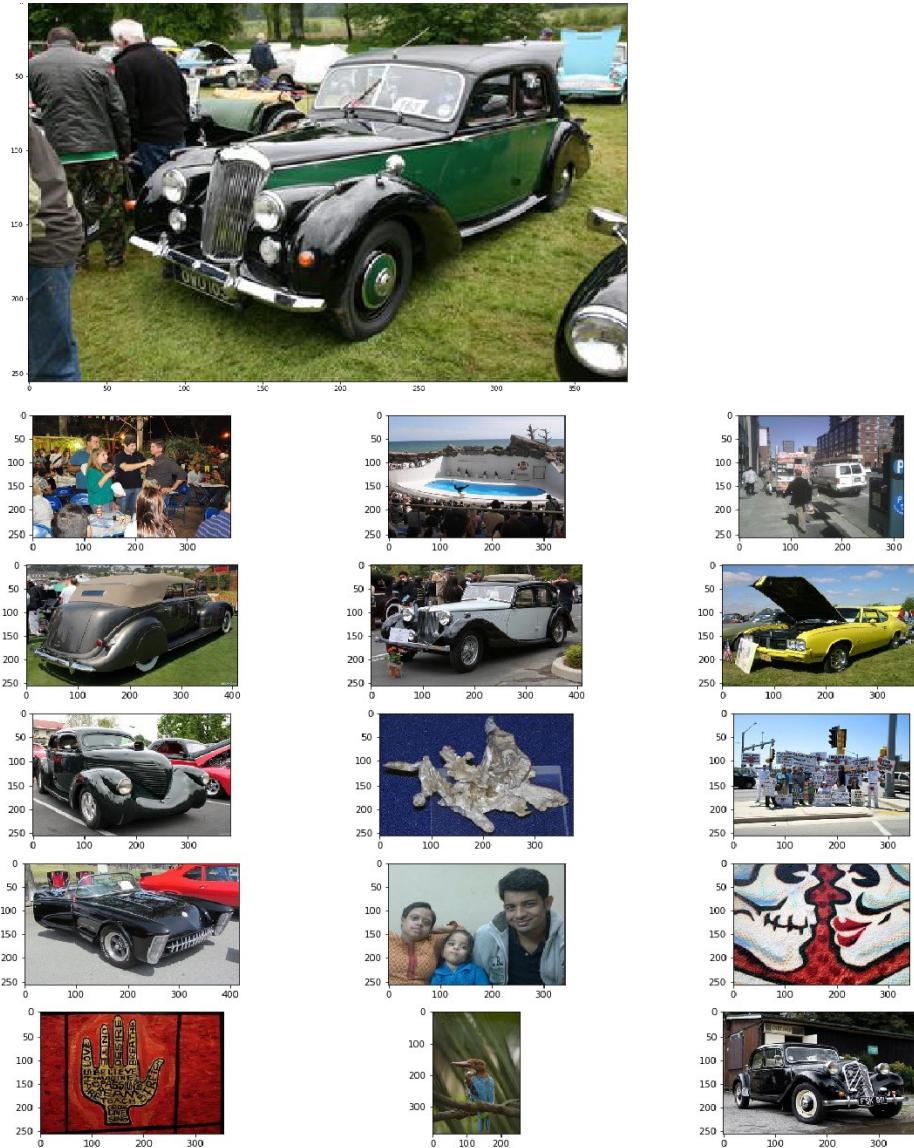
Slika 3.4: Slika na temelju koje se rade preporuka (lijevo) i slika u top 5 koja nije ispravna (desno)

Dobri rezultati sažimanja, kao i lažni pozitivi dobiveni korištenjem jednostavnog *VGG* modela, u dalnjim testiranjima koristili smo kao referencu uspješnosti ekstrakcije značajki iz ulazne slike kod komplikiranijih modela.

3.3. Predtrenirana arhitektura Inceptionv4

Sljedeća arhitektura sa značajno većim kapacitetom je *Inception v4*. Kao i u slučaju arhitekture *VGG* i za *Inception v4* smo preuzeli težinske koeficijente iz arhitekture naučene na skupu ImageNet.

Vođeni zaključcima koje smo dobili testirajući arhitekturu *VGG* za dohvaćanje sličnih slika, prvi način ekstrakcije značajki, kao i u arhitekturi *VGG*, bio je uzimanje izlaza zadnjeg sloja konvolucijske mreže. U ovom slučaju to je vektor od 1536 elemenata dobiven globalnim sažimanjem prosječnom vrijednošću. Dobiveni vektori su se kasnije, korištenjem *PCA* modela, reducirali na 300 elemenata što je dalo loše rezultate unatoč velikom kapacitetu mreže (Slika 3.5).



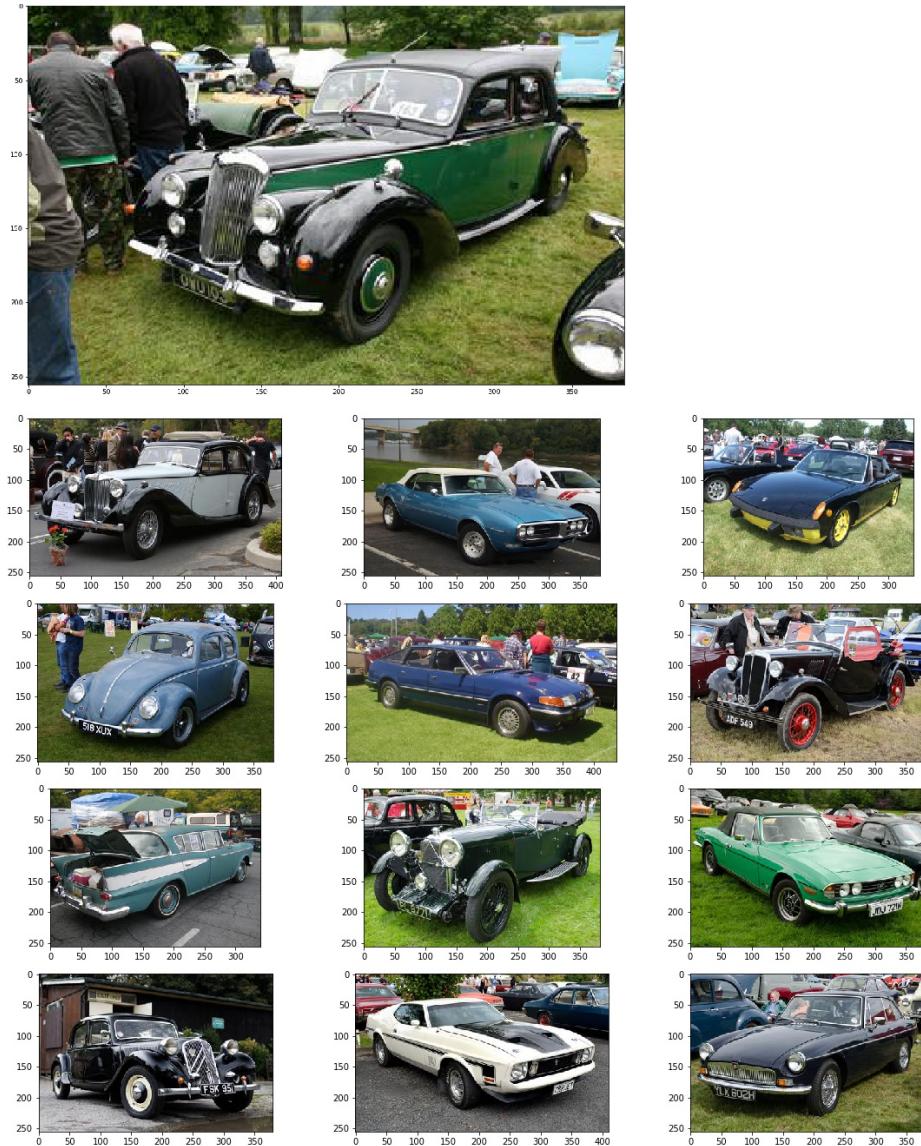
Slika 3.5: Top 12 sličnih rezultata za sliku na vrhu

Razlog zašto redukcija predzadnjeg sloja dovodi do ovoliko loših rezultata djelomično se može objasniti time što mreža taj sloj koristi za završnu klasifikaciju i time nema neku semantičku interpretaciju. Budući da je krajnji cilj dobiti vektor koji se sastoji od 300 elemenata gdje je intenzitet svakog elementa u vektoru označava pojavljivanje određenog uzorka na ulaznoj slici, odlučili smo značajke ekstrahirati kao maksimalni odziv svakog konvolucijskog sloja.

Ekstrakcija značajki se vrši se na način da se nad svakim konvolucijskim slojem u *Inception* mreži napravi operacija globalnog sažimanja. Na primjer, ako je izlaz konvolucijskog sloja $12 \times 12 \times 512$ (visina, širina, broj kanala), globalnim sažimanjem svakog od kanala dobiti ćemo vektor $1 \times 1 \times 512$. Dobiveni vektori svakog konvolucijskog sloja nakon globalnog sažimanja se izravnaju i konkateniraju u jedan veliki vektor s ~ 16000 elemenata. Tako do-

biveni elementi se zatim reduciraju na vektore od 300 elemenata korištenjem *PCA* modela. Kako bi rezultati bili što precizniji, korišteni *PCA* model je prethodno istreniran korišteњem 100.000 ekstrahiranih vektora nasumično odabranih slika radi preciznijeg određivanja glavnih komponenti podatkovnog skupa.

Primjer tako dobivenih rezultata može se vidjeti na sljedećim slikama 3.6, 3.7, 3.7, a korišteni kod za generiranje kao i dodatne rezultate moguće je vidjeti na [25].



Slika 3.6: Top 12 sličnih rezultata za sliku na vrhu



Slika 3.7: Top 12 sličnih rezultata za sliku na vrhu



Slika 3.8: Top 12 sličnih rezultata za sliku na vrhu

Kao što se vidi iz prikazanih slika moguće je reprezentirati sliku sa svega 300 elemenata što predstavlja višestruku uštedu prostora (3.1) pri čemu vektor i dalje dobro ekstrahira sruštinu slike te slične grupira u bliske nakupine. To svojstvo je posebno vidljivo u sljedećem poglavljju.

Tablica 3.1: Primjer uštede prostora

Dimenzije slike	Ukupno elemenata	Faktor uštede
480p (12:9)	640x480	921 600
720p Widescreen	1280x720	2 764 800
1080p HD Widescreen	1920x1080	6 220 800

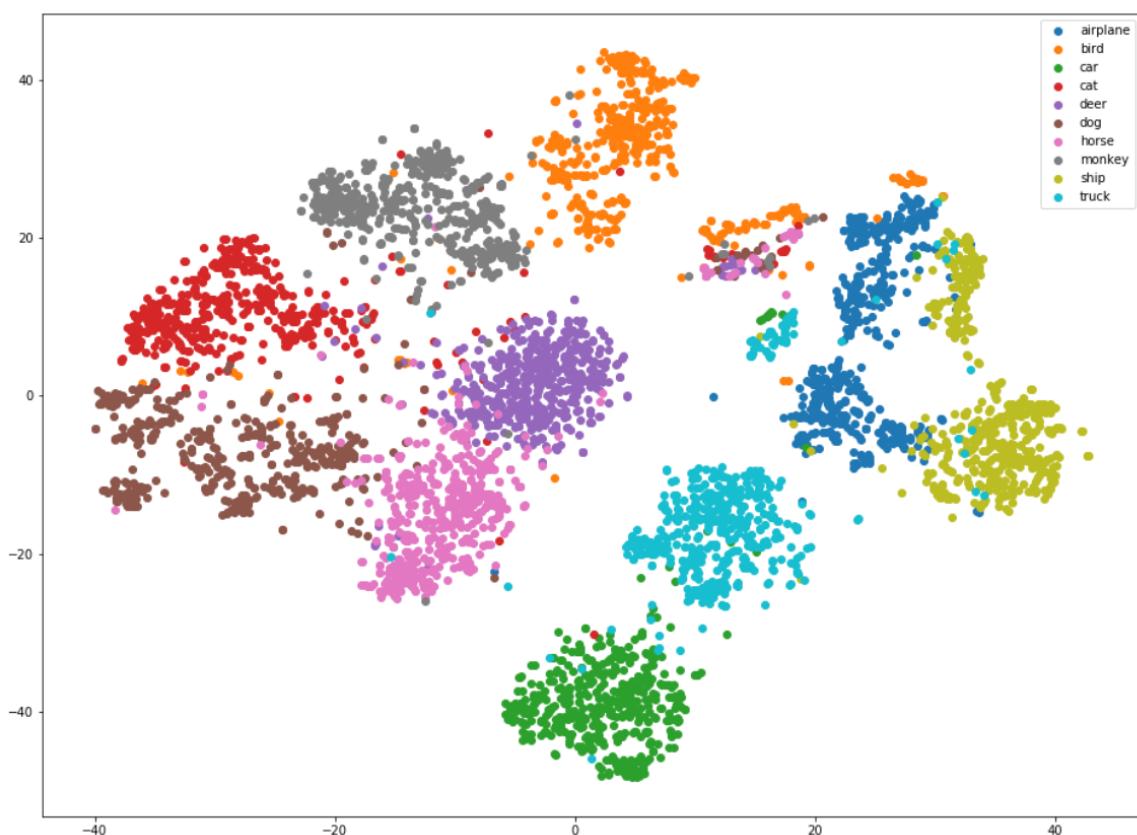
3.4. T-SNE prikaz smanjenog skupa podataka

Budući da je korišteni podskup slika *Google*-ovog *OpenImage* [26] podatkovnog skupa velik (~1 milijun slika) i anotacije klase nisu nužno točne jer ih je većina generirana automatskom strojnom anotacijom korištenjem servisa poput *Google Cloud Vision API*, za vizualizaciju smo odabrali manji skup podataka koji je anotiran ručno i precizno.

Odabrani skup podataka je *STL-10* [27] čije slike imaju dimenziju 96x96x3, a nastao je po uzoru na podatkovni skup CIFAR-10 koji se obično koristi za vrednovanje modela.

Ekstrahirane značajke, dobivene korištenjem najboljeg modela *Inception v4* s globalnim sažimanjem kanala, su pomoću metode *t-SNE* prikazane u 2D prostoru. Na slici 3.9 se vidi da su podaci grupirani u nakupine uz poneka miješanja grupa poput slika jelena i konja. Razlog tome može biti vrlo slična pozadine slike i kut snimanja gdje se značajke jelena (rogovi) ne vide jasno na slici zbog čega ih mreža lako može zamijeniti zbog približno jednake boje i građe.

Moguće je dodatno poboljšati performanse predtrenirane mreže za određivanje sličnosti tako da se glavne komponente *PCA* modela biraju samo na značajkama dobivenim iz klasa koje se žele preciznije razlikovati, u ovome slučaju klase iz *STL-10* podatkovnog skupa.



Slika 3.9: STL-10 dataset reprezentativni vektori prikazani u dvije dimenzije

4. Demo preporučitelj po sadržaju

Kako bismo demonstrirali ponašanje razvijanog sustava za pronalaženje sličnih slika u sklopu projekta implementirana je i jednostavna *web* aplikacija.

Aplikacija je inspirirana sličnim proizvodom, društvenom mrežom temeljenom na dijeljenju slika pod nazivom Pinterest, koja organizira podatke poput slika i video zapisa s cijelog interneta [28].

Naš sustav karakterizira puno manji volumen podataka s kojim rukujemo, ali koncept ostaje isti: prepoznati preference korisnika na temelju njegovih interakcija sa sustavom i pronalazak relevantnih podataka kako bi se poboljšalo korisničko iskustvo.

Aplikacija omogućuje pregledavanje *STL-10* skupa podataka gdje korisnik može izdvojiti skup slika od interesa na temelju kojih će sustav isfiltrirati sve slike sličnog konteksta.

Demo je implementiran kao *web* aplikacija kako bismo sustav učinili što pristupačnijim novim korisnicima i postavljen je na *Heroku* platformu¹.

4.1. Tijek rada

Ovaj odjeljak opisuje ponašanje sustava iz korisničke perspektive i kroz niz instrukcija daje pregled mogućnosti preporučitelja. Posjećivanjem navedene *web* stranice aplikacije korisnik se usmjerava na obrazac za prijavu. Uz pretpostavku da korisnik nema profil za pristup stranici, potrebno je napraviti registraciju putem gumba 'Register' koji se nalazi u gornjem desnom kutu stranice. Pri dolasku na stranicu od korisnika se traži unos korisničkog imena, e-maila i lozinke te se po pritisku na 'Sign Up' gumb završava proces registracije (Slika 4.1). Ako su podaci ispravno uneseni, korisnik će biti vraćen na početnu stranicu gdje može iskoristiti kreirani račun za prijavu (Slika 4.2).

Nakon što se uneseni podaci verificiraju korisnik se usmjerava na početnu stranicu gdje može pročitati kratke upute o korištenju aplikacije. Iste se upute u više detalja opisuju u nastavku.

Zajednička komponenta svim stranicama je navigacijski izbornik na vrhu stranice na kojem se nalaze poveznice: "Recommender demo", "Random Images", "Liked Images", "Re-

¹Izvršnoj verziji aplikacije moguće je pristupiti na linku <https://recommender-demo.herokuapp.com>.

The screenshot shows a registration form titled "Recommender demo". At the top right are "Register" and "Login" links. The form has three input fields: "Username", "Email", and "Password", each with a corresponding label above it. Below the password field is a "Sign Up" button.

Slika 4.1: Stranica za registraciju novog korisnika

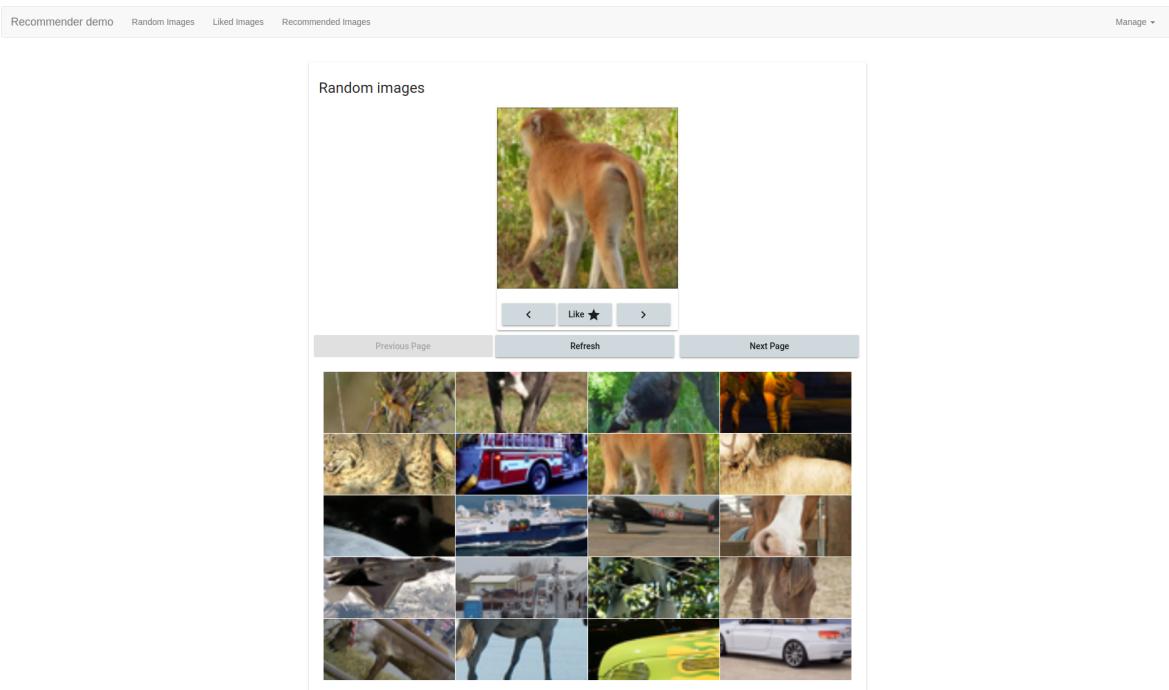
The screenshot shows a login form titled "Recommender demo". At the top right are "Register" and "Login" links. The form has two input fields: "Username" and "Password", each with a corresponding label above it. Below the password field is a "Sign In" button.

Slika 4.2: Stranica za unos korisničkih podataka

commended Images" te "Manage" izbornik s opcijom odjave "Log Out". "Recommender demo" poveznica je početnu stranicu, "Random Images" vodi na galeriju svih slika u sustavu, "Liked Images" poveznica je na galeriju slika koje je korisnik označio kao favorite dok je "Recommended Images" poveznica na galeriju slika koje sustav preporučuje na temelju korisničkih preferencija.

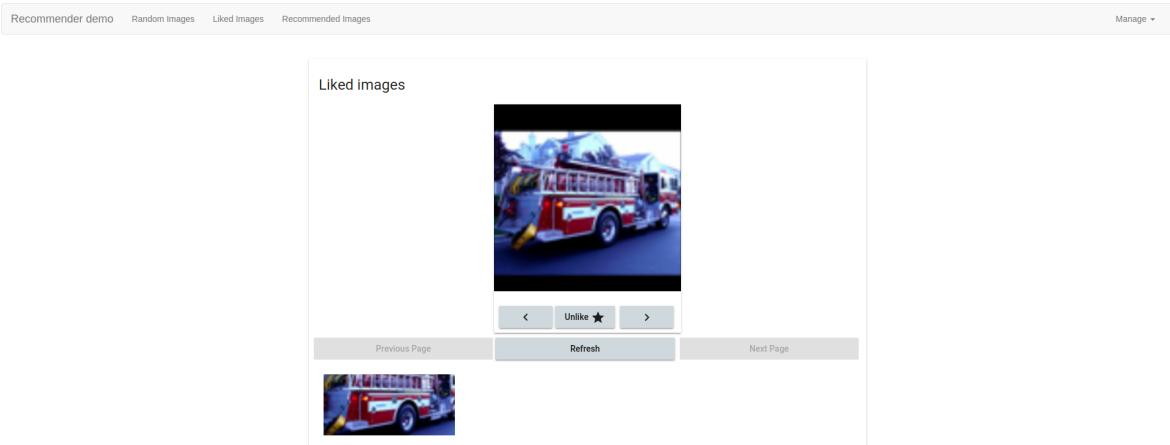
Prvi korak u korištenju aplikacije je odabir slika u galeriji nasumičnih slika korištenjem poveznice "Random Images"(Slika 4.3). Ako korisnik odluči posjetiti neku od preostalih galerija prije nego što je označio ikakvu sliku, dobit će poruku upozorenja da sustav nije mogao

obraditi njegov zahtjev. Jednom kad se galerija učitala, korisnik može početi pregledavati i označavati ponuđene slike. Sučelje se sastoji od dvije glavne komponente: trenutno razmatrane slike i podskupa slika galerije u obliku albuma. Album trenutnog pogleda na slike sustava pruža brzi pregled trenutne stranice slika. Gumbima "Previous Page" i "Next Page" korisnik može dohvaćati preostale stranice i postepeno pretražiti cijelu bazu slika. Gumbom "Refresh" korisnik može zatražiti osvježavanje trenutne stranice slika. Pritiskom na bilo koju od slika u albumu, slika se fokusira u gornjem pregledniku. Ispod fokusirane slike nalaze se tri gumba: gumb za pomak na prethodnu sliku albuma, gumb "Like" za dodavanje slike u favorite i gumb za pomak na iduću sliku albuma. Od korisnika se očekuje odabir najmanje jedne slike i pritisak na gumb "Like" prije prelaska u iduću fazu aplikacije.



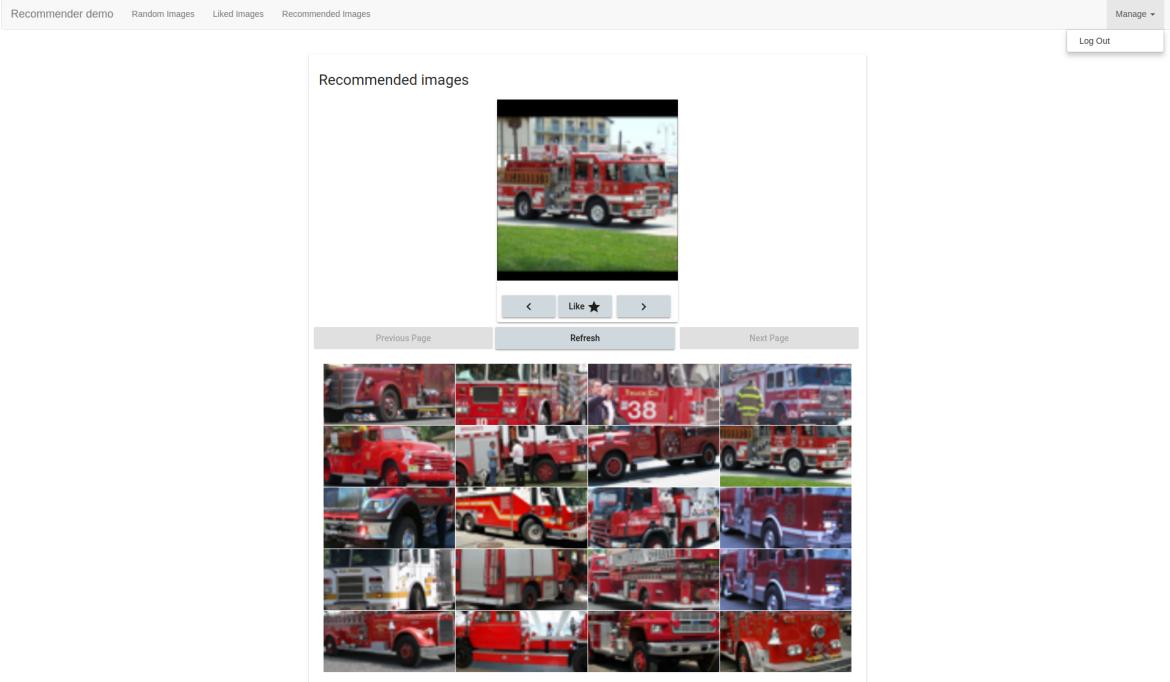
Slika 4.3: Galerija svih slika

Nakon što je korisnik odabrao njemu interesantne slike, može ih još jednom pregledati pritiskom na poveznicu "Liked Images" (Slika 4.4). Kroz ovaj pogled korisnik također može izbrisati neke od slika iz svojih favorita kako bi promotrio kako će to utjecati na preporuke sustava.



Slika 4.4: Galerija odabralih korisničkih slika

Posljednja faza korištenja je pregledavanje preporučenih slika (Slika 4.5). Slično kao i u prethodnim pogledima, korisnik može dodavati slike u favorite, uz iznimku da sada pritiskom na "Refresh" gumb može brže zatražiti novi set preporuka prilagođen ažuriranom stanju preferencija.



Slika 4.5: Galerija preporučenih slika

U nastavku slijedi par primjera odabralih ulaza i rezultirajućih preporuka. Sve rezultate moguće je reproducirati kroz *web* stranicu.

Odabriom prve slike ptice koja se nudi u albumu i odlaskom na pregled preporučenih slika možemo se uvjeriti da su dohvaćene slike uglavnom manjih ptica, ptica sličnih šara ili kombinacije navedenih svojstava (Slika 4.6).



Slika 4.6: Primjer preporuka za sliku ptice

Ako sada izbrišemo preference i odaberemo drugačiju sliku poput purana na slici 4.7, preporuke se mijenjaju u slike većih ptica dugačkih vratova i nogu ili slika ptica slične posture poput našeg referentnog primjera.



Slika 4.7: Primjer preporuka za sliku purana

Provđimo još jedno testiranje ponovno uključujući sliku iz prvog eksperimenta(Slika 4.8). Sada možemo pronaći primjere koji su se pojavili i u prijašnjim upitima, ali i neke nove rezultate. Po završetku korištenja korisnik je dužan odjaviti se pritiskom na "Manage" te "Log Out" gumb u gornjem desnom uglu.



Slika 4.8: Primjer preporuka za kombinaciju slika ptice i purana

4.2. Implementacijski detalji

Kao što je bilo naznačeno u poglavlju o paketima, implementacija *web* poslužitelja oslanja se na dva programska jezika: *Python* i *JavaScript*. Poslužitelj² je pisan u *Python*-u, oslanjajući se na *Django* radno okruženje kako bi se podaci mogli dostavljati korisniku. Podaci aplikacije pohranjeni su u *PostgreSQL* bazi podataka na samom serveru, dok se slike poslužuju s *Amazon S3* servisa. Klijent³ je razvijen u standardnim *web* tehnologijama *HTML* (engl. *HyperText Markup Language*), *CSS* (engl. *Cascading Style Sheets*) i *JavaScript*-u. Kako bi kod bio čim jednostavniji za razumijevanje i implementaciju, odabran je *Angular 5* radni okvir za lakše oblikovanje komponenti klijenta. Po završetku razvoja aplikacija je postavljena na *Heroku* platformu koja omogućuje izgradnju, pokretanje i upravljanje aplikacija. Budući da implementacija same *web* aplikacije nije bila predmet ovog rada zainteresirani čitatelji mogu dobiti više informacija posjećivanjem poveznica u podnožju stranice koje vode na repozitorije koda aplikacije.

²Kod poslužitelja dostupan je na *Github* repozitoriju kojem možete pristupiti na linku <https://github.com/vribic/recommender-demo-api>.

³Kod klijenta dostupan je na *Github* repozitoriju kojem možete pristupiti na linku <https://github.com/vribic/recommender-demo>.

5. Zaključak

Rezultati su pokazali da je korištenjem dubokih modela moguće značajno reducirati reprezentaciju slika na nisko dimenzionalan vektor koji čuva njenu semantiku. Takva redukcija ostvaruje veliku uštedu korištene radne memorije i zbog svoje male dimenzionalnosti omogućuje brzu usporedbu i dohvati sličnih slika preko njihovih reduciranih reprezentacija.

Budući da je u okviru ovog istraživanja korišten veliki i raznoliki skup podataka, kao i duboki modeli koji su trenirani na skupu podataka s drugačijom distribucijom od onoga na kojem su testirani, to ostavlja prostor za dodatne eksperimente za koje naši rezultati određuju donju granicu performansi. Razumno je pretpostaviti da bi se dodatna poboljšanja postigla kada bi se duboki model korišten za redukciju istrenirao na skupu podataka s istom distribucijom kao podaci koji će se koristiti u sustavu preporuke. Tako istrenirani model, omogućio bi izbor manje dimenzije reprezentativnog vektora čime bi se ostvarile još bolje performanse sustava u vidu preciznosti i memorijskih/vremenskih zahtjeva pretrage.

Daljnja istraživanja, kao i napreci u području dubokog učenja, ostavljaju mnogo prostora za detaljnije istraživanje dodatnih mogućnosti ekstrahiranja značajki.

Ekstrahiranje značajki slika pomoću dubokog učenja u svrhu poboljšanja sustava preporuke slika

Sažetak

Slike čine veliki udio ukupnog volumena podataka na internetu. Zbog svojeg bogatog sadržaja koje pohranjuju, sve više suvremenih sustava ima potrebu za efikasnim upravljanjem i analizom slika. Stoga je fokus našeg rada na isprobavanju novih metoda iz područja analize slika kako bi se doskočilo novonastalim zahtjevima. Rad problemu pristupa kroz metodologiju dubokog učenja koja je posljednjih godinama privukla puno pažnje akademije i industrije pri radu s podacima bogatih informacijama, poput slika, teksta ili zvuka. Rezultat rada je sustav koji za dani set slika stvara nisko dimenzionalni zapis koji zadržava semantiku slike i pogodan je za usporedbu temeljem poznatih matematičkih metrika sličnosti. Pri testiranju različitih pristupa isprobani su predtrenirani modeli konvolucijskih neuronskih mreža, metode za redukciju dimenzionalnosti i metrike usporedbe. Kao primjer korištenja razvijenog sustava u sklopu ovog rada implementiran je sustav za preporučivanje slika na temelju korisničkih preferenci.

Ključne riječi: Sustav za preporučivanje, Analiza slika, Duboko učenje, Konvolucijske neuronske mreže, Redukcija dimenzionalnosti, Pretraga velikih skupova podataka

Extracting Image Features Using Deep Learning for Better Image Recommendation System Performance

Abstract

Images make up a large share of the total volume of data on the Internet. Because of their information rich content there exists a large demand for efficient handling and analysis of this data. Therefore our work focuses on testing novel methods from the field of image analysis in order to tackle the problems described. Due to recent advances in the field of deep learning there is an increasing interest from both the academia and industry in domains with huge datasets, such as images, text or audio. The outcome of this project is a functioning system which for a given set of images generates a low dimensional representation suitable for similarity search using existing mathematical tools. While designing the system we looked into different architectures and pretrained models of convolutional neural networks, methods for dimensionality reduction and similarity metrics. As a proof of concept we implemented an image recommendation system based on user preferences.

Keywords: Recommendation systems, Image analysis, Deep learning, Convolutional neural networks, Dimensionality reduction, Information retrieval in big data

LITERATURA

- [1] Fei-Fei Li, Andrej Karpathy, and Justin Johnson. Cs231n: Convolutional neural networks for visual recognition 2016. URL <http://cs231n.stanford.edu/>.
- [2] Siniša Šegvić. Neslužbene stranice predmeta duboko učenje. URL <http://www.zemris.fer.hr/~ssegvic/du/>.
- [3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [4] Jonathon Shlens. A tutorial on principal component analysis. *CoRR*, abs/1404.1100, 2014. URL <http://arxiv.org/abs/1404.1100>.
- [5] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. URL <http://arxiv.org/abs/1409.1556>.
- [6] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016. URL <http://arxiv.org/abs/1602.07261>.
- [7] A. Heidarian and M. J. Dinneen. A hybrid geometric approach for measuring similarity level among documents and document clustering. In *2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 142–151, March 2016. doi: 10.1109/BigDataService.2016.14.
- [8] How many digital photos will be taken in 2017. <https://mylio.com/true-stories/tech-today/how-many-digital-photos-will-be-taken-2017-repost>. Accessed: 2018-04-15.
- [9] Number of internet users worldwide from 2005 to 2017. <https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/>. Accessed: 2018-04-15.
- [10] Internet trends report 2017. <http://www.kpcb.com/file/2017-internet-trends-report>. Slide: 35, Accessed: 2018-04-15.

- [11] Yin Hua Han Dezhi Yu Fei Hu, YuPing. The application of similar image retrieval in electronic commerce. *The Scientific World Journal*, 2014:7, 2014.
- [12] T M Lehmann, M O Güld, Christian Thies, B Fischer, K Spitzer, Daniel Keysers, Hermann Ney, M Kohnen, H Schubert, and Berthold Wein. Content-based image retrieval in medical applications. 43:354–61, 02 2004.
- [13] Eriksson Monteiro, Frederico Valente, Carlos Costa, and José Oliveira. A recommender system for medical imaging diagnostic. 210:461–3, 05 2015.
- [14] Thomas Joseph. Smart city analysis using spatial data and predicting the sustainability. *CoRR*, abs/1406.4986, 2014. URL <http://arxiv.org/abs/1406.4986>.
- [15] ElizaYingzi Du, Robert Ives, Alan van Nevel, and Jin-Hua She. Advanced image processing for defense and security applications. *EURASIP Journal on Advances in Signal Processing*, 2010(1):432972, Mar 2011. ISSN 1687-6180. doi: 10.1155/2010/432972. URL <https://doi.org/10.1155/2010/432972>.
- [16] Y. Chen, V. Roussev, G. Richard, and Y. Gao. Content-based image retrieval for digital forensics. In Mark Pollitt and Sujeeet Shenoi, editors, *Advances in Digital Forensics*, pages 271–282, Boston, MA, 2005. Springer US. ISBN 978-0-387-31163-0.
- [17] 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, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion 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, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <https://www.tensorflow.org/>. Software available from tensorflow.org.
- [18] Itseez. Open source computer vision library. <https://github.com/itseez/opencv>, 2015.
- [19] Leonid Boytsov and Bilegsaikhan Naidan. Engineering efficient and effective non-metric space library. In *Similarity Search and Applications - 6th International Conference, SISAP 2013, A Coruña, Spain, October 2-4, 2013, Proceedings*, pages 280–293, 2013. doi: 10.1007/978-3-642-41062-8_28. URL https://doi.org/10.1007/978-3-642-41062-8_28.

- [20] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. URL <http://www.jmlr.org/papers/v9/vandermaaten08a.html>.
- [21] Lawrence Journal-World. Django. <https://github.com/django/django>, 2005.
- [22] Google. Angular. <https://github.com/angular/angular>, 2016.
- [23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and 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.
- [24] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). URL <http://www.cs.toronto.edu/~kriz/cifar.html>.
- [25] Toni Vlaić, Viran Ribić, and Lucija Šikić. Extracting deep features for image recommendation. <https://github.com/Mungosin/AVSP>, 2016.
- [26] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2017.
- [27] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL <http://proceedings.mlr.press/v15/coates11a.html>.
- [28] Wikipedia contributors. Pinterest — Wikipedia, the free encyclopedia, 2018. URL <https://en.wikipedia.org/w/index.php?title=Pinterest&oldid=836282502>. [Online; accessed 16-April-2018].