Generativno modeliranje uvjetnim normalizirajućim tokovima

Ćosić, Tomislav

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 596

GENERATIVNO MODELIRANJE UVJETNIM NORMALIZIRAJUĆIM TOKOVIMA

Tomislav Ćosić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 596

GENERATIVNO MODELIRANJE UVJETNIM NORMALIZIRAJUĆIM TOKOVIMA

Tomislav Ćosić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 596

Pristupnik:	**Tomislav Ćosić (0036526583)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	prof. dr. sc. Siniša Šegvić

Zadatak: **Generativno modeliranje uvjetnim normalizirajućim tokovima**

Opis zadatka:

Procjena gustoće i generiranje slika važni su zadatci računalnog vida s mnogim zanimljivim primjenama. Poznato je da generativni modeli slika ne uspijevaju naučiti semantičke koncepte. Ovaj problem možemo ublažiti uvjetovanjem generativnog modela semantičkim informacijama. U okviru rada, potrebno je odabrati okvir za automatsku diferencijaciju te upoznati biblioteke za rukovanje tenzorima i slikama. Proučiti i ukratko opisati postojeće generativne arhitekture. Odabrati slobodno dostupne skupove slika te oblikovati podskupove za učenje, validaciju i testiranje. Oblikovati uvjetni generativni tok te uhodati postupke učenja i validiranja hiperparametara. Primijeniti naučene modele, prikazati eksperimente na javno dostupnim podatcima te usporediti generalizacijsku izvedbu sa stanjem tehnike. Komentirati učinkovitost učenja i zaključivanja. Predložiti pravce za budući rad. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 28. lipnja 2024.

Zahvaljujem mentoru prof. dr. sc. Siniši Šegviću i Anji Delić, univ. mag. ing. na strpljenju i pomoći u izradi ovog rada. Hvala mojoj obitelji, Dori i prijateljima na podršci kroz cijelo školovanje.

Sadržaj

1. Uvod

Veliki pozitivni pomak u stanju tehnike na cijelom nizu problema koje je vrlo teško definirati kroz okvir tradicionalnih algoritamskih postupaka dogodio se sredinom 2010-ih godina. Uzrok za taj pomak primarno možemo tražiti u napretcima u području računalnih komponenti i njihove računske moći, ponajviše grafičkih kartica. Posljedično, duboko učenje se pokazalo novom najboljom opcijom za mnoštvo područja, među kojima se mi usredotočujemo na obradu slike.

Konkretno, koristimo konvolucijske modele za generiranje korisnih reprezentacija iz slike i normalizirajuće tokove za evaluaciju izglednosti slike, i posljedično, detekciju anomalija.

U ovom radu napravit ćemo pregled dosadašnjeg rada u ovom području i korištenih metoda. Zatim ćemo napraviti pregled teoretske pozadine ekstrakcije vektora značajki iz ulaznih slika, normalizirajućih tokova i korištenja normalizirajućih tokova za detekciju anomalija. Predstavit ćemo provedene eksperimente i njihove rezultate. Razmotrit ćemo dobivene rezultate i diskutirati o dobivenim rezultatima. Konačno, izvući ćemo zaključke i predložiti neke zanimljive smjerove za buduća istraživanja u ovom području.

1.1. Detekcija anomalija u kontekstu modernih primjena dubokog učenja

Detekcija anomalija je zanimljiv zadatak u kontekstu modernih primjena dubokih modela. Naime, u laboratorijskom radu često svjesno ili nesvjesno unosimo određena ograničenja na sustave koje razmatramo. Primjerice, može se raditi o razmatranju isključivo zatvorenog skupa podataka s konačnim brojem klasa i neizbježno ograničenim brojem slika koje mogu biti ulaz u model koji razmatramo. U takvom scenariju znamo da ulaz

ne može biti bilo kakav, to jest znamo da će objekt na ulaznoj slici pripadati jednoj od klasa iz skupa podataka. Nadalje, razmatrajući performanse modela kroz dobivene rezultate eksperimenata, lako je zanemariti određene greške koje model učini. Naravno, nije realno očekivati savršene performanse modela, no s druge strane treba uzeti u obzir da primjerice u primjenama kao što su autonomna vozila, pojedine greške mogu biti neprihvatljive, čak i uz globalno vrlo dobre performanse modela.

Takve primjene nameću sve veću važnost ne samo točnosti modela, nego i robusnosti. Detekcija anomalija ovdje može imati ključnu ulogu jer omogućuje detekciju situacija u kojima se u sceni događa nešto neobično na što model nije dobro prilagođen, umjesto da model u pokušaju postizanja što boljih rezultata s relativno velikom mjerom pouzdanosti donosi odluke u situacijama za koje nije dobro pripremljen kroz proces učenja.

2. Normalizirajući tokovi

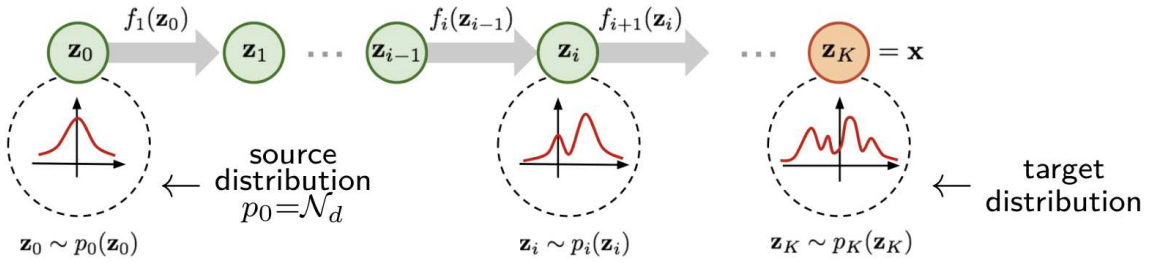
Kako smo spomenuli u uvodnom poglavlju, za evaluaciju izglednosti ulazne slike koristimo normalizirajuće tokove [1]. Radi se o vrsti generativnih modela koji, za razliku od nekih drugih često korištenih generativnih modela kao što su varijacijski autoenkoderi [2, 3] ili generativne suprotstavljene mreže (GAN) [4], imaju mogućnost egzaktne evaluacije log-izglednosti podataka [5]. To svojstvo normalizirajućih tokova nam je značajno u kontekstu detekcija anomalija, gdje se nadamo iskoristiti pretpostavku da će evaluirana izglednost slika koje sadrže anomalije biti manja u odnosu na evaluiranu izglednost slika koje ju ne sadrže.

2.1. Princip rada normalizirajućih tokova

Normalizirajući tokovi su generativni modeli. Cilj im je nepoznatu distribuciju p_D skupa podataka D nad kojim uče modelirati distribucijom p_ϕ, gdje je ϕ skup parametara modela. Konkretnije, parametre modela čine parametri pojedinačnih transformacija. Normalizirajući tokovi sastoje se od niza transformacija kojima se slika postupno transformira iz točke u distribuciji skupa podataka, preko reprezentacija slike u latentnim distribucijama nakon svake od transformacija, sve do reprezentacije slike u odabranoj (time i nama poznatoj) osnovnoj distribuciji (slika 2.1.). Za osnovnu distribuciju, tipično odabiremo normalnu distribuciju.

2.1.1. Transformacije i izglednost

Sljedeće jednadžbe prikazuju kako transformacijama prelazimo od nama nepoznate distribucije podataka p i konkretnog podatka iz te distribucije x do konačne odabrane osnovne distribucije q, gdje je z_K reprezentacija podatka x u toj distribuciji q. Do te reprezentacije z_K smo došli nakon K transformacija nad počevši od x, od distribucije podataka



Slika 2.1. Normalizirajući tokovi nizom naučenih transformacija prelaze iz osnovne distribucije u ciljanu distribuciju podataka. Izvor: [6]

p prema distribuciji q.

$$p(\mathbf{x}) = q(\mathbf{z}_K)\left|\det\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right| \tag{2.1}$$

$$p(\mathbf{x}) = q(\mathbf{z}_K)\left|\det\prod_{k=1}^{K}\frac{\partial \mathbf{z}_k}{\partial \mathbf{z}_{k-1}}\right|, \mathbf{z}_k = f_k(\mathbf{z}_{k-1}), \mathbf{z}_0 = \mathbf{x} \tag{2.2}$$

$$p(\mathbf{x}) = q(\mathbf{z}_K)\left|\prod_{k=1}^{K}\det\frac{\partial \mathbf{z}_k}{\partial \mathbf{z}_{k-1}}\right|, \mathbf{z}_k = f_k(\mathbf{z}_{k-1}), \mathbf{z}_0 = \mathbf{x} \tag{2.3}$$

$$p(\mathbf{x}) = q(\mathbf{z}_K)\prod_{k=1}^{K}\left|\det\frac{\partial \mathbf{z}_k}{\partial \mathbf{z}_{k-1}}\right|, \mathbf{z}_k = f_k(\mathbf{z}_{k-1}), \mathbf{z}_0 = \mathbf{x} \tag{2.4}$$

$$f = f_1 \circ f_2 \circ f_3 \circ \ldots \circ f_K \tag{2.5}$$

$$\int_{-\infty}^{\infty} f(x)\,dx = 1 \tag{2.6}$$

$$\int_{a}^{b} f(x)\,dx = 1 \tag{2.7}$$

Jednadžba 2.1 nam kaže da distribuciju p možemo prikazati kao distribuciju q koja je prošla kroz transformaciju f, gdje s f označavamo kompoziciju pojedinačnih transformacija (jednadžba 2.5). Uloga množenja s apsolutnom vrijednosti Jakobijana je korek-

cija volumena, pošto znamo da postoji ograničenje na funkciju gustoće vjerojatnosti da mora vrijediti jednadžba 2.6, odnosno njezin ekvivalent 2.7 u slučaju ograničene domene na intervalu $[a, b]$.

Nadalje, transformaciju opisanu u jednadžbi 2.1 raspisujemo u jednadžbi 2.2, navodeći pojedinačne transformacije uz pravilo ulančavanja za parcijalne derivacije.

U jednadžbi 2.3, nad jednadžbom 2.2 primjenjujemo Binet-Cauchyjev teorem koji kaže da je determinanta umnoška dviju kvadratnih matrica istog reda jednaka umnošku determinanti svake matrice.

U jednadžbi 2.4, nad jednadžbom 2.3 iskorištavamo činjenicu da je apsolutna vrijednost umnoška dva realna broja jednaka umnošku apsolutnih vrijednosti: $|a \cdot b| = |a| \cdot |b|$.

Također, umjesto izraza za izglednost 2.4, u praksi nam je korisniji izraz za log-izglednost 2.8 Razlog je što uzastopno množenje u 2.4, može dovesti do vrijednosti koje su premale ili prevelike da bi se ispravno zapisale u memoriju računala (*engl. numerical underflow / overflow*).

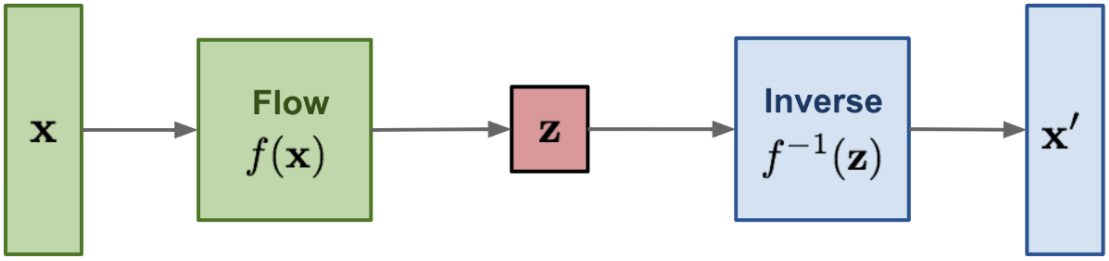
U smislu optimizacije, cilj ostaje isti jer će isti skup parametara θ maksimizirati izglednost neovisno o primjeni logaritma pošto logaritam ima svojstvo monotonosti.

$$\ln p_\theta(\mathbf{x}) = \ln q(\mathbf{z}_K) + \sum_{k=1}^{K} \ln \left| \det \frac{\partial \mathbf{z}_k}{\partial \mathbf{z}_{k-1}} \right|, \quad \mathbf{z}_k = f_{\theta_k}(\mathbf{z}_{k-1}), \mathbf{z}_0 = \mathbf{x} \tag{2.8}$$

2.1.2. Bijektivnost

Važna karakteristika transformacija u normalizirajućim tokovima je da moraju biti bijektivne. To nas donekle ograničava u oblikovanju i odabiru transformacija, ali nam omogućava da na naučenom normalizirajućem toku generiramo nove podatke iz naučene distribucije tako da prvo uzorkujemo poznatu distribuciju, a zatim nad tim uzorkom primijenimo transformacije u smjeru od poznate distribucije prema distribuciji „stvarnih" podataka koju smo modelirali. Taj je proces prikazan na slici 2.2.

Primjena slijeda transformacija nad slikom x rezultira vrijednosti z u prostoru osnovne distribucije. Primjena inverznih transformacija f^{-1} nad z rezultira podatkom x'



Slika 2.2. Stvaranje sintetičkih podataka primjenom inverza naučene transformacije. Preuzeto s [7]

u prostoru originalne slike. S obzirom da je f^{-1} inverz transformacije f, x' je jednak x. U praksi, postoje male razlike između x i x' zbog nepreciznosti zapisa decimalnih brojeva u računalu.

2.2. Uvjetni normalizirajući tokovi

Dalje gradeći na izloženim principima normalizirajućih tokova, od interesa su nam uvjetni normalizirajući tokovi. Oni nam omogućuju da kod evaluacije izglednosti u jednom smjeru toka i kod generiranja podataka u drugom smjeru toka dodamo uvjetnu informaciju. U našem slučaju, uvjetni tok će biti uvjetovan klasom primjera. To znači da možemo evaluirati uvjetnu izglednost $p(x|c)$ gdje su x slika, a c klasa slike. Također kod generiranja slike, možemo generirati sliku odabrane klase.

Kako bismo to postigli, potrebno je izmijeniti osnovnu distribuciju ili arhitekturu modela. Tu ćemo razmotriti dva pristupa za modeliranje skupa podataka s K klasa: korištenje K normalizirajućih tokova ili korištenje jednog toka s osnovnom distribucijom s K komponenti.

U slučaju korištenja K normalizirajućih tokova, svaki tok ima zadaću modelirati distribuciju slika jedne od klasa. Sam tok, njegove transformacije i osnovna distribucija, mogu ostati onakvima kako smo prikazali u općenitom pregledu normalizirajućih tokova.

U slučaju jednog toka s osnovnom distribucijom s K komponenti, uobičajenu normalnu osnovnu distribuciju zamjenjujemo osnovnom distribucijom koja je multimodalna, s K komponenti. Primjerice, možemo koristiti model Gaussove mješavine. Onda

svaki od modova distribucije modelira funkciju gustoće vjerojatnosti za jednu klasu iz skupa podataka.

Pristup sa K tokova ima veći kapacitet jer svaka klasa dobiva zaseban tok s vlastitim skupom parametara, dok se kod drugog pristupa isti tok koristi za sve podatke neovisno o klasi. No, veća ekspresivnost dolazi s cijenom da se svaki tok mora odvojeno naučiti.

Također, komplicira se evaluacija izglednosti. Kod pristupa s jednim tokom, prolazom kroz transformacije toka iz prostora slike prema osnovnoj distribuciji, dobivamo jednu reprezentaciju slike u tom prostoru z. Zatim, za klasu c nad odgovarajućim modom osnovne distribucije možemo evaluirati izglednost $p(z|c)$. U slučaju da nas zanima $p(z)$, tu vrijednost možemo dobiti marginalizacijom po klasama, odnosno modovima distribucije (jednadžba 2.9),

$$p(\mathbf{z}) = \sum_c p(\mathbf{z}|c) \tag{2.9}$$

S druge strane, kod evaluacije izglednosti u pristupu s K tokova prvo je potrebno provesti odvojen prolaz kroz tok podatka x u svakom od K tokova (izraz 2.10).

$$\mathbf{z}_c = f_{\theta_c}^{-1}(\mathbf{x}) \tag{2.10}$$

Zatim u svakoj od K osnovnih distribucija evaluiramo izglednost odgovarajućeg z_c. Kako bismo ovdje dobili izglednost $p(z)$, opet ju je moguće dobiti marginalizacijom (izraz 2.11), ali imajući na umu potrebnu korekciju volumena pošto sada imamo K valjanih funkcija gustoće vjerojatnosti, u smislu da svaka ima integral jednak 1.

$$p(\mathbf{z}) = \frac{1}{K} \sum_c p(\mathbf{z}|c) \tag{2.11}$$

U ovom radu, koristili smo pristup s jednim uvjetnim normalizirajućim tokom, gdje je osnovna distribucija model Gaussove mješavine s brojem komponenti jednakim broju klasa. Eventualna potreba za većim kapacitetom modela (što se može smatrati manom u odnosu na drugi prikazani pristup) može se postići transformacijama s većom širinom

za one vrste transformacija u kojima je to moguće ili dužim slijedom transformacija u toku.

3. Ekstrakcija značajki

Kao ulaze u normalizirajući tok, koristimo vektore značajki koji su ugrađivanja (*engl. embedding*) samih slika, to jest korisne reprezentacije slika iz skupa podataka. Prije samog učenja normalizirajućeg toka, potrebno je provesti proces učenja na modelu koji će od slika načiniti njihove reprezentacije. Okvir korišten za ekstrakciju značajki je SimCLR [8].

3.1. SimCLR

SimCLR (A **Sim**ple Framework for **C**ontrastive **L**earning of Visual **R**epresentations) je radni okvir za samonadzirano učenje vizualnih reprezentacija slike pomoći kontrastivnog učenja. Osnovna ideja ovog pristupa je maksimizirati sličnost različito perturbiranih reprezentacija istih podataka, kao i u isto vrijeme minimizirati sličnost različito perturbiranih reprezentacija međusobno različitih podataka. Par koji čine dvije različite reprezentacije istog podatka nazivamo pozitivnim parom, dok dvije različito perturbirane reprezentacije različitih podataka nazivamo negativnim parom. Metoda se sastoji od nekoliko glavnih komponenti: perturbiranja podataka, enkodera (okosnice), projekcijske glave i kontrastivnog gubitka.

3.1.1. Perturbacija podataka

Podatke je potrebno perturbirate kako bismo dobili dvije različite reprezentacije istog ulaznog podatka. Zatim je cilj postići da model stvara slične reprezentacije za takve pozitivne parove, a različite reprezentacije za negativne parove (parove podataka nastale perturbiranjem različitih ulaznih podataka). Perturbacije korištene u ovom radu su:

- nasumično izrezivanje slike

- nasumično horizontalno zrcaljenje slike (s vjerojatnošću 50%)
- distorzija boja na slikama
- pretvaranje slike iz slike u boji u crno-bijelu sliku (s vjerojatnošću 20%).

3.1.2. Enkoder (okosnica)

Enkoder ili okosnica (*engl. backbone*) je model nad kojim se primjenjuje SimCLR metoda u smislu da enkoder od perturbiranih slika radi njihove reprezentacije u obliku vektora značajki te se na temelju kontrastivnog gubitka učenje provodi tako što se ažuriraju parametri enkodera. Kao enkoder u ovom smo radu koristili konvolucijski model ResNet-18 [9]. Za neku ulaznu sliku x, i neka je $f(\cdot)$ enkoder, a neka slike $\tilde{x}_i$ i $\tilde{x}_j$ čine pozitivni par nastao perturbacijom slike x. Izlazi iz enkodera h_i i h_j tada su dani izrazima 3.1, odnosno 3.2

$$h_i = f(\tilde{x}_i) \tag{3.1}$$

$$h_j = f(\tilde{x}_j) \tag{3.2}$$

3.1.3. Projekcijska glava

Izlazi iz enkodera h_i i h_j zatim ulaze u projekcijsku glavu. Uloga projekcijske glave je reprezentacije h_i i h_j projicirati u latentni prostor gdje će se nad njima evaluirati kontrastivni gubitak. U našem radu, kao projekcijsku glavu koristimo jednostavni model koji se sastoji od dva potpuno povezana sloja između kojih se nalazi ReLU aktivacijska funkcija. Za neku reprezentaciju na izlazu enkodera h_i, označimo sa z_i projekciju reprezentacije u latentnom prostoru. z_i tada je dana izrazom 3.3 $W^{(1)}$ i $W^{(2)}$ su matrice težina potpuno povezanih slojeva.

$$z_i = W^{(2)}\sigma\left(W^{(1)}h_i\right) \tag{3.3}$$

$$\sigma(x) = max(0, x) \tag{3.4}$$

3.1.4. Kontrastivni gubitak

Kako smo već naveli, uloga kontrastivnog gubitka je da se okosnica nauči tako da perturbacije iste ulazne slike rezultiraju sličnim latentnim reprezentacijama nakon ekstrahiranja značajki, a reprezentacije slika s različitim objektima rezultiraju i što različitijim reprezentacijama.

Takav optimizacijski cilj postižemo funkcijama gubitka koje će poprimiti male vrijednosti ukoliko su reprezentacije pozitivnih parova slične, a reprezentacije negativnih parova međusobno različite. Analogno, poželjno je da funkcija gubitka poprimi velike iznose kada su reprezentacije pozitivnih parova međusobno više različite, kao i kad su reprezentacije negativnih parova međusobno slične.

Kao kontrastivni gubitak, SimCLR koristi NT-Xent [10] (*engl. Normalized Temperature-Scaled Cross Entropy*). Autori SimCLR-a pokazali su da je za metodu SimCLR NT-Xent gubitak bolji izbor od alternativa kao što su logistički gubitak (*engl. logistic loss*) [11] ili gubitka s marginom [12] (*engl. margin loss*).

Za N podataka u grupi, okosnica stvara $2N$ reprezentacija. Detaljnije, za danu sliku x, okosnica generira dvije reprezentacije z_i i z_j uz različito perturbiranje slike x te prolaz kroz okosnicu. z_j je onda pozitivni par reprezentacije z_i, a preostalih $2(N-1)$ reprezentacija koje su nastale iz ostalih slika u grupi čine negativne parove promatrane reprezentacije z_i.

Iznos doprinosa gubitka za neki par reprezentacija dan je izrazom 3.5, gdje sličnost sama dva vektora značajki evaluiramo kosinusnom sličnosti (izraz 3.6), a $[\![\cdot]\!]$ je oznaka za Iversonove zagrade.

$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} [\![k \neq i]\!] \exp(\text{sim}(z_i, z_k)/\tau)} \tag{3.5}$$

$$\text{sim}(z_i, z_j) = \frac{z_i \cdot z_j}{||z_i||||z_j||} \tag{3.6}$$

U izrazu 3.5, τ označava temperaturu. Temperatura je hiperparametar koji utječe na koncentriranost distribucije vrijednosti sličnosti, to jest koliki utjecaj promjena u izračunatoj sličnosti ima na iznos gubitka. Naime, kad je τ malen, eksponencijalna funkcija (i posljedično sam gubitak) postaje osjetljivija na razlike u izračunatim vrijednostima sličnosti. S druge strane, velike vrijednosti τ imaju utjecaj da različite izračunate sličnosti nakon eksponenciranja uzrokuju manju promjenu iznosa gubitka. Stoga, male vrijednosti hiperparametra τ potiču model da se više usredotoči na najsličnije pozitivne parove i na najmanje slične negativne parove, što može voditi ka boljim diskriminativnim svojstvima. S druge strane, prednost većih vrijednosti τ je što zaglađuju optimizacijski krajolik, čime mogu pomoći u sprječavanju zaglavljivanja u lokalnom optimumu. Također, smanjivanje hiperparametra τ uzrokuje povećanje magnitude gradijenata tijekom unazadnog prolaza kroz okosnicu tijekom učenja.

$$\mathcal{L} = \frac{1}{2N} \sum_{i=1}^{N} (\ell_{i,2i-1} + \ell_{i,2i}) \tag{3.7}$$

Konačno, ukupni gubitak 3.7 čini aritmetička sredina izraza 3.5 evaluiranim nad svim pozitivnim parovima u grupi (za svaki pozitivni par se evaluacija 3.5 izračunava za obje perturbirane reprezentacije, to jest oba člana para).

4. ResNet-18

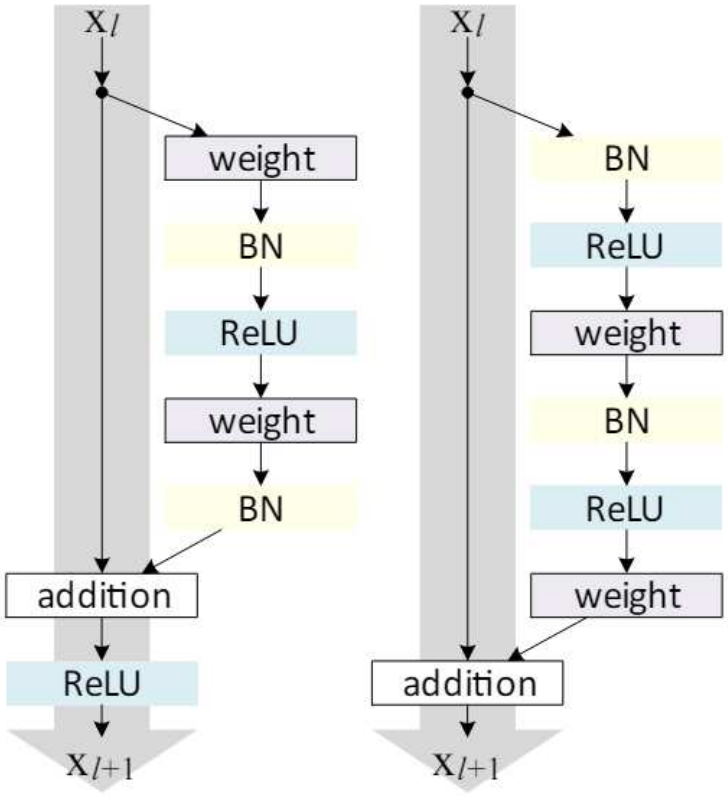
ResNet [9] (skraćeno od *engl.* ***Res****idual* ***Net****work*) je obitelj dubokih konvolucijskih rezidualnih modela ponajviše korištenih na slikama. Glavni doprinos ovih modela je uvođenje rezidualnih veza. One omogućuju učinkovito učenje dubljih modela nego do tada, suočavajući se s problemom nestajućeg gradijenta kod unazadnog prolaza kroz model tijekom učenja.

U ovom radu koristimo model ResNet-18. ResNet-18 se sastoji od 18 slojeva, među kojima su konvolucijski slojevi, slojevi normalizacije po grupi (*batch normalization*), aktivacijske funkcije (ReLU), kao i potpuno povezani slojevi.

ResNet se primarno koristi za klasifikaciju pa stoga među zadnjim slojevima modela imamo sažimanje prosjekom (*average pool*), „izravnavanje" značajki u vektor, aktivacijska funkcija te potpuno povezani sloj čija je uloga dobivanje logita duljine jednake broju klasa. Pošto ćemo u ovom radu ResNet-18 koristiti kao okosnicu u SimCLR-u, njegova uloga neće biti klasifikacija nego generiranje vektora značajki iz ulazne perturbirane slike. Stoga je potrebno prenamijeniti model za novu primjenu određenim zahvatima na arhitekturi modela. Uklanjamo posljedni potpuni povezani sloj te, ovisno o eksperimentu, uklanjamo ranije spomenutu aktivacijsku funkciju koja se nalazi prije posljednjeg povezanog sloja. To znači da su vektori značajki koje koristimo zapravo predlogiti modela, dobiveni nakon posljednjeg sloja sažimanja u modelu.

Također, za potrebe ekstrakcije značajki koristimo unaprijeđenu verziju ResNeta-18 [13]. Razlika je prikazana na slici 4.1. U [13] se navodi da je pomicanje aktivacijske funkcije prije zbrajanja s residualnim vezama korisno zbog lakše optimizacije i boljeg korištenja normalizacije po grupi kao regularizacijskog mehanizma. Ipak, nama je primarni interes izbjeći da se sve negativne vrijednosti u vektoru značajki na izlazu iz sloja

postave na nulu.



Slika 4.1. Originalni ResNet blok lijevo i PreAct ResNet blok desno. Pomicanje aktivacijske funkcije koja više nije posljednja operacija nad tenzorom je poželjno u kontekstu ekstrakcije značajki. Iz rada [13]

.

5. Mjere za evaluaciju anomalnosti

Mjere navedene u potpoglavljima ispod korištene su tijekom evaluacije uvjetnog normalizirajućeg toka kako bismo dobili kvantitativne rezultate o sposobnosti modela na zadatku detekcije anomalija.

5.1. FPR@95

FPR@95 označava stopu lažnih pozitiva (*engl. False Positive Rate*, FPR, izraz 5.2) kada je stopa stvarnih pozitiva (*engl. True Positive Rate*, TPR, izraz 5.1) 95%. Stopa stvarnih pozitiva odgovara odzivu.

U kontekstu detekcije anomalija, stvarni pozitiv je podatak koji nije anomalija i model je procijenio da se ne radi o anomaliji, a lažni pozitiv je anomalija za koju je model pogrešno ocijenio da se ne radi o anomaliji. Manje vrijednosti ove mjere označavaju bolje performanse modela, označavajući da je manje primjera koji su anomalije pogrešno označeno kao primjeri koji nisu anomalija.

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{5.1}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{5.2}$$

5.2. AUROC

AUROC (*engl. **A**rea **U**nder the **R**eceiver **O**perating **C**haracteristic Curve*) mjera predstavlja skalarnu vrijednost koja u našoj primjena na detekciji anomalija sažima performanse

detekcije anomalija modela preko svih pragova. Pragom se ovdje smatra granica izglednosti ispod koje se dani podatak smatra anomalijom.

Koristeći ranije navedene mjere stope stvarnih pozitiva TPR (izraz 5.1) i stope lažnih pozitiva (izraz 5.2), AUROC dobivamo po izrazu 5.3

$$\text{AUROC} = \int_0^1 \text{TPR}(\text{FPR})\, d(\text{FPR}) \tag{5.3}$$

AUROC predstavlja izglednost da je nasumično odabrani primjer koji nije anomalija izgledniji od nasumično odabranog primjera koji je anomalija. AUROC jednak 1 označavao bi savršene performanse razlikovanja anomalija od primjera koji nisu anomalija, dok AUROC od 0.5 predstavlja performanse na razini nasumičnog pogađanja.

5.3. AUPR-IN

AUPR-IN mjera (***A**rea **U**nder the **P**recision-**R**ecall Curve for **In**-Distribution*) mjeri površinu ispod krivulje preciznosti i odziva (*engl. precision-recall curve*) kada pozitivnom klasom smatramo primjere koji nisu anomalije. U načelu, ova mjera daje indikaciju o performansama modela kada treba za primjere odrediti da ne pripadaju anomalijama.

Za potrebe izračuna, definiramo mjeru preciznosti (izraz 5.4).

$$\text{P} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5.4}$$

Također, definiramo mjeru odziva (izraz 5.5).

$$\text{R} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5.5}$$

AUPR-IN tada računamo po izrazu 5.6

$$\text{AUPR-IN} = \int_0^1 \text{P}(\text{R})\, d\text{R} \tag{5.6}$$

5.4. AUPR-OUT

Analogno AUPR-IN, mjera AUPR-OUT (***A**rea **U**nder the **P**recision-**R**ecall Curve for **Out**-of-Distribution*) mjeri površinu ispod krivulje preciznosti i odziva (*engl. precision-recall curve*) kada su nam od interesa primjeri koji su anomalije. Ova mjera daje indikaciju o performansama modela kada za primjere koji su anomalije model detektirati da se radi o anomalijama.

S obzirom na to da se ovaj puta u primjerima radi o anomalijama, drukčije definiramo preciznost (izraz 5.7).

$$\mathrm{P} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}} \tag{5.7}$$

Također, drukčije definiramo i mjeru odziva (izraz 5.8).

$$\mathrm{R} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{5.8}$$

Konačno, AUPR-OUT tada možemo izračunati po izrazu 5.9

$$\text{AUPR-OUT} = \int_0^1 \mathrm{P}(\mathrm{R})\, d\mathrm{R} \tag{5.9}$$

6. Skupovi podataka

Sljedeći skupovi podataka korišteni su za učenje modela i za evaluaciju performansi modela na zadatku detekcije anomalija.

6.1. ImageNet

ImageNet [14] je skup slika nastao po uzoru na leksički skup podataka WordNet [15]. U svom punom obliku, trenutno se sastoji od više od 14 000 000 slika koje pripadaju jednoj od više od 20 000 klasa.

Često se koristi i verzija ImageNeta nastala za ILSVRC [16] (**I**mageNet **L**arge **S**cale **V**isual **R**ecognition **C**hallenge) natjecanje, tamo korištena od 2012. do 2017. godine. Ova verzija se često naziva i ImageNet-1K jer sadrži slike iz 1000 klasa.

U ovom radu, u nekima od eksperimenata koristimo SimCLR okosnicu predtreniranu na ImageNet skupu podataka.

6.2. CIFAR

CIFAR [17] je skup slika nastao označavanjem dijela slika iz skupa *80 million tiny images* [18]. Sastoji se od 60 000 slika i dolazi u dvije verzije: CIFAR-10, sa 10 klasa i 6000 slika po klasi, i CIFAR-100, sa 100 klasa i 600 slika po klasi. Dodatno, tih 100 klasa je grupirano u 20 nadklasa.

6.3. Tiny ImageNet (TIN)

Tiny ImageNet [19] je skup slika koji se sastoji od 100 000 slika, po 500 u svakoj od 200 klasa. Nastao je kao alternativa CIFAR skupovima podataka smanjivanjem slika iz Ima-

Slika 6.1. Primjeri slika iz ImageNeta. Iz rada [14].

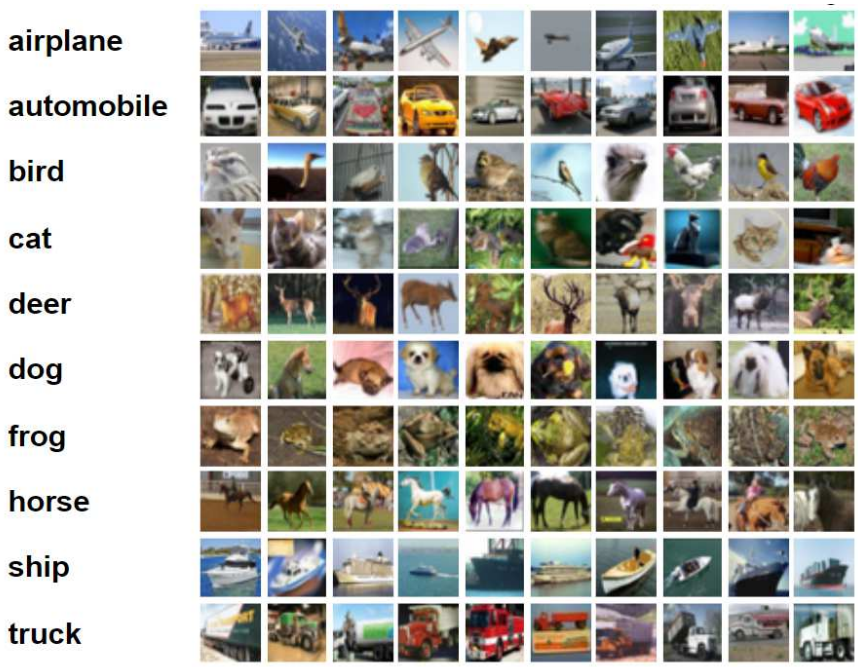
geNeta na veličinu 64x64 piksela.

6.4. MNIST

MNIST [20] je skup koji se sastoji od 60 000 slika rukom pisanih znamenki. Slike su crno-bijele, dimenzija 28x28 piksela,

6.5. SVHN (Street View House Numbers)

SVHN [21] je skup podataka dobiven izrezivanjem slika kućnih brojeva iz snimki *Google Street Viewa* [22]. Stilom se može smatrati sličnim MNIST-u jer se radi o slikama znamenki. Sastoji se od preko 600 000 slika.



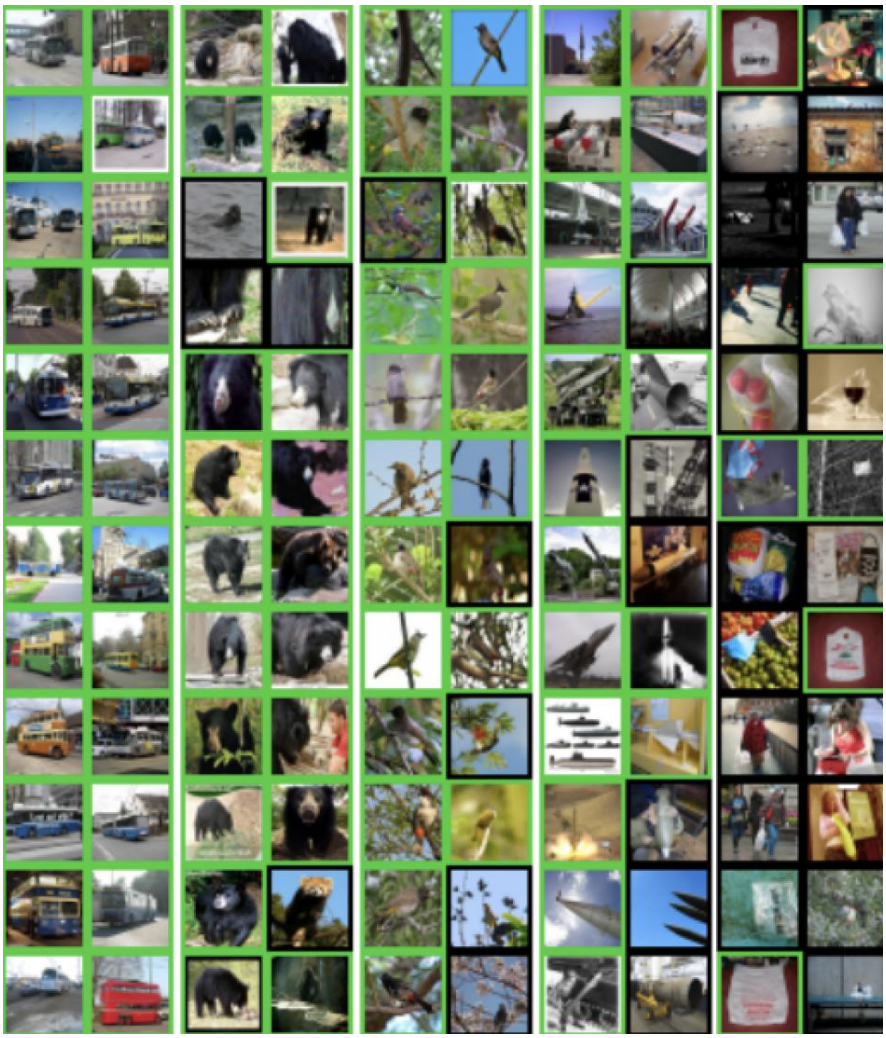
Slika 6.2. Primjeri slika iz skupa CIFAR-10. Iz rada [17].

6.6. Describable Textures Dataset (DTD)

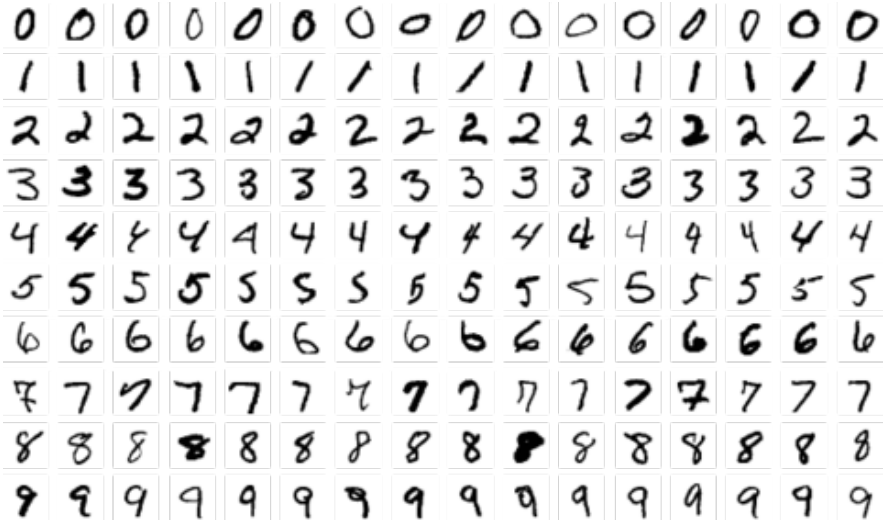
Describable Textures Dataset [23] je skup podataka koji se sastoji od 5640 slika raznih tekstura smještenih u 47 klasa inspiriranih ljudskom percepcijom tekstura i ploha (primjerice „naborano" ili „točkasto").

6.7. Places365

Places365 [24] je skup podataka koji se sastoji od preko 10 milijuna slika smještenih u više od 400 scena.Skup sadrži između 5000 i 30 000 slika po klasi, konzistentno s učestalosti pojavljivanja određenih scena u stvarnome svijetu.



Slika 6.3. Primjeri slika iz skupa Tiny ImageNet. Iz rada [19].



Slika 6.4. Primjeri slika iz skupa MNIST.

Slika 6.5. Primjeri slika iz skupa SVHN.

Slika 6.6. Primjeri slika iz skupa Describable Textures Dataset.

Slika 6.7. Primjeri slika iz skupa Places365.

7. Eksperimenti

SimCLR metodom učimo okosnicu. Uloga okosnice je kasnije na ulazu dobiti sliku i na izlazu generirati sažetu reprezentaciju ulazne slike u obliku vektora značajki. Težine naučene okosnice i generirane značajke po potrebi spremamo u datoteke za kasniju upotrebu.

Zatim, uvjetni normalizirajući tok učimo na pripremljenim skupovima generiranih reprezentacija slika. Evaluacija izglednosti primjera nakon prolaza kroz uvjetni normalizirajući tok nam služi za odluku je li primjer anomalan. Dobivena vrijenost izglednosti se uspoređuje sa nizom pragova kako bi se dobila odluka je li primjer anomalan. Tako dobivamo podatke potrebne za računanje metrika u poglavlju 8.

7.1. ReLU

U eksperimentima isprobavamo dodati i ukloniti prolaz kroz aktivacijsku funkciju ReLU na samom kraju unaprijednog prolaza kroz okosnicu. To nam je zanimljivo jer taj prolaz efektivno uklanja pola signala iz reprezentacija (sve vrijednosti u vektorima značajki manje ili jednake nuli postavljaju se na nulu). Zanimljivo je razmotriti radi li se o značajnom gubitku informacije, to jest hoće li i koliko detekcija anomalije biti manje uspješna.

7.2. Prednaučena okosnica

Također, u eksperimentima istražujemo koliki je utjecaj na kvalitetu generiranih reprezentacija slike ovisno polazimo li u učenju okosnice od ResNeta prednaučenog na skupu podataka ImageNet. Zanimljivo je razmotriti hoće li prednaučena okosnica generirati kvalitetnije vektore značajki i posljedično tako pomoći uvjetnom toku na zadatku detekcije anomalija.

7.3. Korišteni alati

7.3.1. Python i PyTorch

Programski kod pisan je u programskom jeziku Python [25]. Za razvoj modela, korišten je radni okvir PyTorch [26].

7.3.2. SimCLR radni okvir

Za učenje okosnice, korištena je javno dostupna implementacija [27] ranije opisane metode SimCLR.

7.3.3. OpenOOD

Za učenje uvjetnog normalizirajućeg toka i evaluaciju performansi modela na zadatku detekcije anomalija, korištena je dopunjena implementacija alata OpenOOD [28, 29].

Tijekom evaluacije performansi, razmatramo evaluacije performansi modela na dvije varijante anomalija, NearOOD i FarOOD.

NearOOD odnosi se na anomalije koje su bliže distribuciji podataka koji nisu anomalije, dok se FarOOD odnosi na anomalije značajnije drukčije distribucije. Primjerice, za skup podataka koji se sastoji od slika pasa, NearOOD anomalija bi mogla biti slika mačke ili lisice. To nisu slike psa, ali i mačke i lisice imaju vizualne sličnosti psima. FarOOD anomalija mogla bi biti slika automobila ili rukom pisane znamenke.

8. Rezultati

U sljedećim tablicama navodimo dobivene rezultate eksperimenata na zadatku evaluacije anomalija gdje kao model koristimo uvjetni normalizirajući tok, a kao generator vektora značajki iz slika koristimo ResNet-18. U redovima tablice se nalaze korišteni skupovi podataka, grupirani po tome jesu li bili korišteni za NearOOD ili FarOOD detekciju anomalija. Također, navodimo i prosječne vrijednosti nad skupovima unutar tih grupa. Stupci označavaju svaku od mjerenih metrika.

Razmatramo četiri scenarija, ovisno postoji li konačni prolaz kroz aktivacijsku funkciju ReLU na kraju generiranja vektora značajki te je li okosnica prednaučena na ImageNet skupu podataka ili ne.

Također, donosimo dvije tablice (8.5. i 8.6.) s usporedbom prosječnih metrika u odnosu na referentni scenarij: detekciju anomalija kad okosnica nije prednaučena i vektori značajki nemaju konačan prolaz kroz ReLU.

Dataset	FPR@95	AUROC	AUPR-IN	AUPR-OUT
Near OOD				
CIFAR100	96.80	48.91	47.82	49.98
TIN	96.71	49.71	51.87	47.20
Mean	96.75	49.31	49.84	48.59
Far OOD				
MNIST	100.00	0.09	5.90	72.16
SVHN	98.91	50.79	23.74	76.91
Texture	97.77	64.53	66.66	60.44
Places365	97.37	47.99	18.96	78.21
Mean	98.51	40.85	28.82	71.93

Tablica 8.1. Performanse toka na zadatku detekcije anomalija (okosnica nije prednaučena, ReLU na kraju ekstrakcije značajki)

Kada okosnica nije prednaučena, a na kraju ekstrakcije značajki imamo ReLU (tablica 8.1.), vidimo da performanse pretežno nisu značajno bolje od nasumičnog pogađa-

nja. Iznimka je prepoznavanje da su primjeri anomalni (AUPR-OUT) kada su anomalije značajno različite od ostalih primjera. Tu model postiže rezultate koji su ipak bolji od nasumičnog pogađanja. Također, tada su i metrike na Texture skupu podataka nešto bolje. S druge strane, neke metrike su izuzetno loše, primjerice FarOOD AUROC na vrlo jednostavnom MNIST skupu podataka.

Dataset	**FPR@95**	**AUROC**	**AUPR-IN**	**AUPR-OUT**
Near OOD				
CIFAR100	94.34	50.36	50.37	50.58
TIN	92.58	53.17	56.68	49.14
Mean	93.46	51.76	53.52	49.86
Far OOD				
MNIST	86.33	61.79	18.67	91.98
SVHN	96.65	43.90	22.67	70.45
Texture	91.61	57.88	67.23	46.96
Places365	93.02	52.93	22.52	81.29
Mean	91.90	54.13	32.77	72.67

Tablica 8.2. Performanse toka na zadatku detekcije anomalija (okosnica je prednaučena na ImageNetu, ReLU na kraju ekstrakcije značajki)

Kada je okosnica prednaučena na ImageNetu i u okosnici postoji ReLU na kraju ekstrakcije značajki, također vidimo slične rezultate kao i u prethodnom scenariju. Zanimljivo je da uz FarOOD i MNIST, rezultati značajno variraju prema poprilično dobrima i poprilično lošima, ovisno o promatranoj metrici.

Dataset	**FPR@95**	**AUROC**	**AUPR-IN**	**AUPR-OUT**
Near OOD				
CIFAR100	100.00	52.53	53.62	52.07
TIN	100.00	54.34	58.41	50.13
Mean	100.00	53.43	56.01	51.10
Far OOD				
MNIST	80.67	64.94	24.82	92.19
SVHN	100.00	47.87	27.26	73.09
Texture	89.60	61.20	71.00	48.80
Places365	100.00	54.48	25.95	81.92
Mean	92.57	57.13	37.26	74.00

Tablica 8.3. Performanse toka na zadatku detekcije anomalija (okosnica je prednaučena na ImageNetu, bez ReLU na kraju ekstrakcije značajki)

U scenariju kada je okosnica prednaučena na ImageNetu i nema ReLU-a na kraju ekstrakcije značajki, model postiže konzistentno slabe rezultate na metrici FPR@95. Također, na FarOOD model postiže ili dobre rezultate na AUPR-IN i loše na AUPR-OUT

ili obrnuto, ovisno o skupu podataka.

Dataset	FPR@95	AUROC	AUPR_IN	AUPR_OUT
Near OOD				
CIFAR100	97.48	48.85	47.45	50.52
TIN	97.17	52.94	53.33	51.81
Mean	97.32	50.89	50.39	51.16
Far OOD				
MNIST	2.32	99.39	98.31	99.88
SVHN	96.71	48.68	23.18	78.65
Texture	99.84	35.45	49.69	37.36
Places365	94.19	55.96	22.83	83.53
Mean	73.26	59.87	48.50	74.85

Tablica 8.4. Performanse toka na zadatku detekcije anomalija (okosnica nije prednaučena, bez ReLU na kraju ekstrakcije značajki)

Kada okosnica nije prednaučena i ne koristi se ReLU na kraju ekstrakcije značajki, rezultati su pretežno usporedivi s nekim od ranijih scenarija, osim kod skupa MNIST i FarOOD evaluaciju, gdje model postiže značajno bolje rezultate.

Scenarij	FPR@95	AUROC	AUPR-IN	AUPR-OUT
Okosnica nije prednaučena, bez ReLU	97.32	50.89	50.39	51.16
Okosnica nije prednaučena, s ReLU	-0.57	-1.58	-0.55	-2.57
Prednaučena okosnica, s ReLU	-3.86	+0.87	+3.13	-1.30
Prednaučena okosnica, bez ReLU	+2.68	+2.54	+5.65	-0.06

Tablica 8.5. Razlike u prosječnim vrijednostima metrika za NearOOD za svaki od scenarija.

Tablica 8.5. prikazuje prosječne vrijednosti za svaku metriku u svakom scenariju kod NearOOD evaluacije. Primjećujemo da korištenje ReLU-a na kraju generiranja vektora značajki ne uzrokuje konzistentna poboljšanja ili pogoršanja.

S druge strane, kad je okosnica prednaučena na ImageNetu, na većini evaluiranih metrika postiže bolje rezultate nego kada nije.

Scenarij	FPR@95	AUROC	AUPR-IN	AUPR-OUT
Okosnica nije prednaučena, bez ReLU	73.26	59.87	48.50	74.85
Okosnica nije prednaučena, s ReLU	+25.25	-19.97	-19.68	-2.08
Prednaučena okosnica, s ReLU	+18.64	-5.74	-15.73	-2.18
Prednaučena okosnica, bez ReLU	+19.31	-2.74	-11.24	-0.85

Tablica 8.6. Razlike u prosječnim vrijednostima metrika za FarOOD za svaki od scenarija.

U tablici 8.6. prikazane su prosječne vrijednosti za svaku metriku u svakom scenariju kod FarOOD evaluacije. Iznenađujuće, kombinacija okosnice koja nije prednaučena

i nekorištenja ReLU-a na kraju procesa ekstrakcije značajki postiže daleko najbolje rezultate. Ipak, prednaučena okosnica s ReLU postiže većinom najgore rezultate, a i u slučaju prednaučene okosnice, rezultati su nešto bolji ukoliko se ne koristi ReLU na kraju ekstrakcije značajki. To je potencijalno indikacija da gubitak informacija u značajkama kod prolaska kroz ReLU ima primjetan negativni utjecaj kada se radi detekcija anomalija gdje su anomalije značajno drukčije od ostalih primjera.

9. Zaključak

U prvom poglavlju smo predstavili zadatak detekcije anomalija i njegovu relevantnost u modernim primjenama i budućnosti.

U sljedećem poglavlju, razmotrili smo princip rada i teoretsku pozadinu normalizirajućih tokova. Usredotočili smo se na uvjetne normalizirajuće tokove, posebno klasom uvjetovane uvjetne tokove koje smo koristili u ovom radu.

Razmotrili smo metodu za učenje modela za generiranje smislenih reprezentacija slika SimCLR. Zatim smo razmotrili ResNet-18, konvolucijski model koji smo učili pomoću SimCLR radnog okvira i koristili kao okosnicu za generiranje reprezentacija na zadatku detekcije anomalija.

Dali smo pregled korištenih mjera za evaluaciju anomalniosti i korištenih skupova podataka.

Opisali smo implementirane eksperimente i predstavili dobivene rezultate. Tu ističemo primjetan prosječno negativan utjecaj ReLU-a na kraju generiranja značajki na sposobnost normalizirajućeg toka u detekciji anomalija. U budućem radu se kroz rezultate nameće niz pojava koje bi bilo zanimljivo istražiti. Primjerice, bilo bi zanimljivo istražiti zašto u nekim scenarijima i nad određenim metrikama tok postiže rezultate koji djeluju lošiji i od nasumičnog pogađanja. S druge strane, ističemo da je tok u nekim scenarijima dao odlične performanse kao detektor anomalija (uz određene kombinacije korištenih metoda i skupa podataka). Bilo bi zanimljivo istražiti što je utjecalo na takav rezultat i može li model biti robustniji, u smislu da je sposoban postizati takve rezultate za veći raspon zadataka.

Nadalje, zanimljivo bi bilo istražiti zašto kod FarOOD evaluacija model često ima

tendenciju ili dobro detektirati da primjeri nisu anomalije ili da jesu, ali se ta pojava ne pojavljuje u NearOOD. Naime, za FarOOD bi jednostavno objašnjenje bilo da model konzistentno daje visoke ili niske vrijednosti izglednosti bez obzira na ulaz, ali onda vjerujemo da bi ova pojava bila prisutna i na NearOOD rezultatima, a čini se da nije.

Literatura

[1] D. J. Rezende i S. Mohamed, “Variational inference with normalizing flows”, 2016.

[2] D. P. Kingma i M. Welling, “Auto-encoding variational bayes”, 2022.

[3] ——, “An introduction to variational autoencoders”, *Foundations and Trends® in Machine Learning*, sv. 12, br. 4, str. 307–392, 2019. https://doi.org/10.1561/2200000056

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, i Y. Bengio, “Generative adversarial networks”, 2014.

[5] I. Kobyzev, S. J. Prince, i M. A. Brubaker, “Normalizing flows: An introduction and review of current methods”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, sv. 43, br. 11, str. 3964–3979, studeni 2021. https://doi.org/10.1109/tpami.2020.2992934

[6] L. Weng, “Flow-based deep generative models”, 2018., accessed: 2024-06-20. [Mrežno]. Adresa: https://lilianweng.github.io/posts/2018-10-13-flow-models/

[7] ——, “Flow-based models”, 2018., accessed: 2024-06-22. [Mrežno]. Adresa: https://lilianweng.github.io/posts/2018-10-13-flow-models/

[8] T. Chen, S. Kornblith, M. Norouzi, i G. Hinton, “A simple framework for contrastive learning of visual representations”, 2020.

[9] K. He, X. Zhang, S. Ren, i J. Sun, “Deep residual learning for image recognition”, 2015.

[10] K. Sohn, “Improved deep metric learning with multi-class n-pair loss objective”, u *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, i R. Garnett, Ur., sv. 29. Curran Associates, Inc., 2016. [Mrežno]. Adresa: https://proceedings.neurips.cc/paper_files/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf

[11] T. Mikolov, K. Chen, G. Corrado, i J. Dean, “Efficient estimation of word representations in vector space”, 2013.

[12] F. Schroff, D. Kalenichenko, i J. Philbin, “Facenet: A unified embedding for face recognition and clustering”, u *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, lipanj 2015. https://doi.org/10.1109/cvpr.2015.7298682

[13] K. He, X. Zhang, S. Ren, i J. Sun, “Identity mappings in deep residual networks”, 2016. [Mrežno]. Adresa: https://arxiv.org/abs/1603.05027

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, i L. Fei-Fei, “Imagenet: A large-scale hierarchical image database”, u *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009., str. 248–255. https://doi.org/10.1109/CVPR.2009.5206848

[15] C. Fellbaum, Ur., *WordNet: An Electronic Lexical Database*, ser. Language, Speech, and Communication. Cambridge, MA: MIT Press, 1998.

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, i L. Fei-Fei, “Imagenet large scale visual recognition challenge”, 2015. [Mrežno]. Adresa: https://arxiv.org/abs/1409.0575

[17] A. Krizhevsky, V. Nair, i G. Hinton, “Cifar-10 (canadian institute for advanced research)”. [Mrežno]. Adresa: http://www.cs.toronto.edu/~kriz/cifar.html

[18] A. Torralba, R. Fergus, i W. T. Freeman, “80 million tiny images: A large data set for nonparametric object and scene recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, sv. 30, br. 11, str. 1958–1970, 2008. https://doi.org/10.1109/TPAMI.2008.128

[19] P. Chrabaszcz, I. Loshchilov, i F. Hutter, “A downsampled variant of imagenet as an alternative to the cifar datasets”, 2017. [Mrežno]. Adresa: https://arxiv.org/abs/1707.08819

[20] L. Deng, “The mnist database of handwritten digit images for machine learning research [best of the web]”, *IEEE Signal Processing Magazine*, sv. 29, br. 6, str. 141–142, 2012. https://doi.org/10.1109/MSP.2012.2211477

[21] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, i A. Y. Ng, “Reading digits in natural images with unsupervised feature learning”, u *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. [Mrežno]. Adresa: http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf

[22] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, i J. Weaver, “Google street view: Capturing the world at street level”, *IEEE Computer*, sv. 43, str. 32–38, 06 2010. https://doi.org/10.1109/MC.2010.170

[23] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , i A. Vedaldi, “Describing textures in the wild”, u *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[24] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, i A. Torralba, “Places: A 10 million image database for scene recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[25] G. Van Rossum i F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, i S. Chintala, “Pytorch: An imperative style, high-performance deep learning library”, 2019. [Mrežno]. Adresa: https://arxiv.org/abs/1912.01703

[27] W. van Heeswijk i contributors, “Simclr: A simple framework for contrastive learning of visual representations”, https://github.com/Spijkervet/SimCLR/, 2021.,

accessed: 2024-06-28.

[28] J. Yang, P. Wang, D. Zou, Z. Zhou, K. Ding, W. Peng, H. Wang, G. Chen, B. Li, Y. Sun, X. Du, K. Zhou, W. Zhang, D. Hendrycks, Y. Li, i Z. Liu, "Openood: Benchmarking generalized out-of-distribution detection", 2022.

[29] J. Zhang, J. Yang, P. Wang, H. Wang, Y. Lin, H. Zhang, Y. Sun, X. Du, K. Zhou, W. Zhang, Y. Li, Z. Liu, Y. Chen, i H. Li, "Openood v1.5: Enhanced benchmark for out-of-distribution detection", *arXiv preprint arXiv:2306.09301*, 2023.

Sažetak

Generativno modeliranje uvjetnim normalizirajućim tokovima

Tomislav Ćosić

Procjena gustoće i generiranje slika važni su zadatci računalnog vida s mnogim zanimljivim primjenama. Poznato je da generativni modeli slika ne uspijevaju naučiti semantičke koncepte. Ovaj problem možemo ublažiti uvjetovanjem generativnog modela semantičkim informacijama. U okviru rada, potrebno je odabrati okvir za automatsku diferencijaciju te upoznati biblioteke za rukovanje tenzorima i slikama. Proučiti i ukratko opisati postojeće generativne arhitekture. Odabrati slobodno dostupne skupove slika te oblikovati podskupove za učenje, validaciju i testiranje. Oblikovati uvjetni generativni tok te uhodati postupke učenja i validiranja hiperparametara. Primijeniti naučene modele, prikazati eksperimente na javno dostupnim podatcima te usporediti generalizacijsku izvedbu sa stanjem tehnike. Komentirati učinkovitost učenja i zaključivanja. Predložiti pravce za budući rad. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Ključne riječi: normalizirajući tok; uvjetni normalizirajući tok; SimCLR; detekcija anomalija; OpenOOD; okosnica

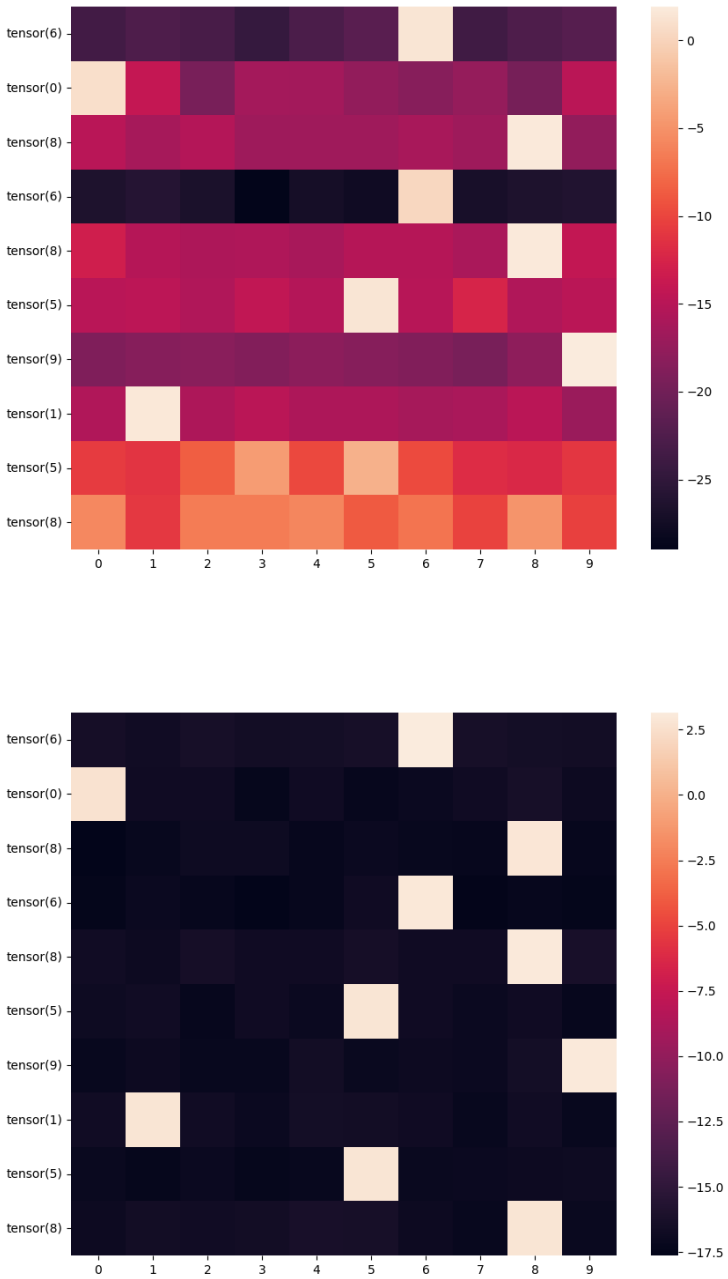
Abstract

Generative modelling using conditional normalizing flows

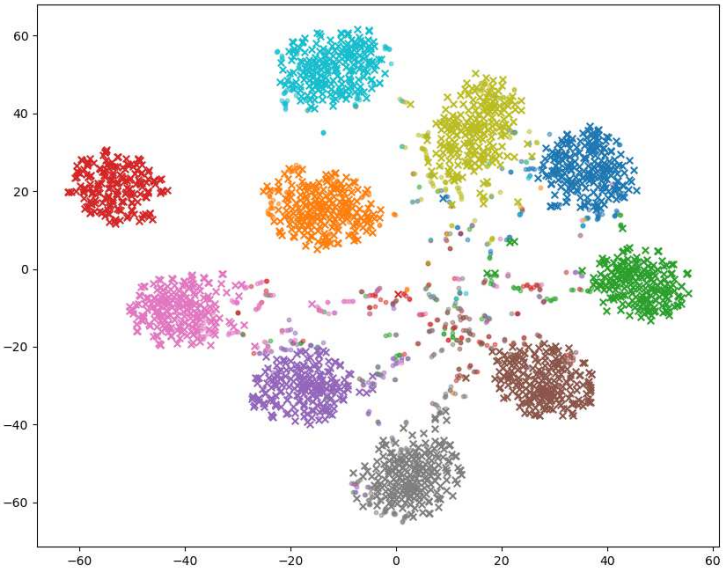
Tomislav Ćosić

Density estimation and image generation are important tasks in computer vision with many interesting applications. It is known that image generative models fail to learn semantic concepts. This problem can be mitigated by conditioning the generative model using semantic information. In the scope of the paper, it is required to choose an automatic differentiation framework and become familiar with libraries used to handle tensors and images. Research and briefly describe existing generative architectures. Choose freely available image dataset and construct subsets for training, validation and testing. Model a conditional generative flow and implement training and hyperparameter validation. Apply trained models, perform experiments on publicly available datasets and compare generative performance with state-of-the-art. Comment on training and inference efficiency. Suggest direction for future work. Attach source code and executable code of developed processes to the paper, experiments and results, along with necessary explanations and documentation. Cite used literature and state received help.

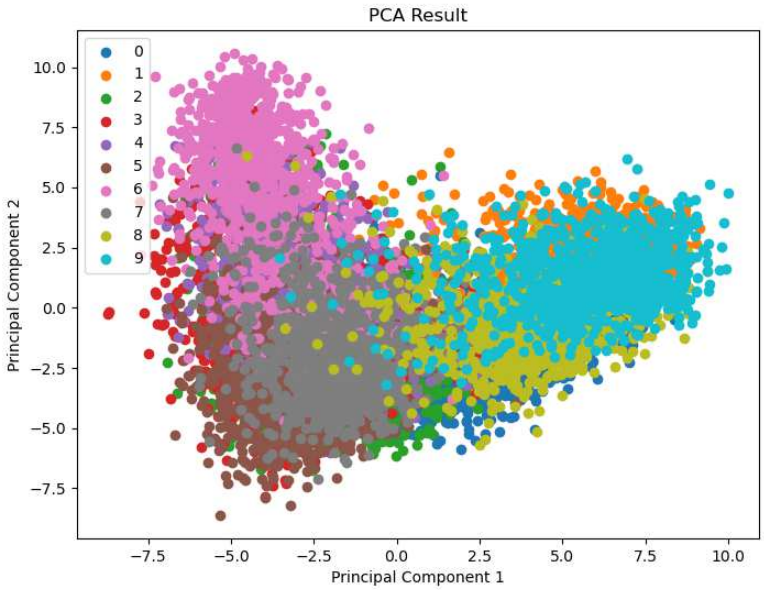
Keywords: normalizing flow; conditional normalizing flow; SimCLR; anomaly detection; OpenOOD; backbone



Slika 9.1. Prikaz vrijednosti izglednosti izabranih primjera po klasama rano i kasno u procesu učenja uvjetnog normalizirajućeg toka.



Slika 9.2. t-SNE analiza reprezentacija slika dobivenih okosnicom naučenom SimCLR-om.



Slika 9.3. PCA reprezentacija slika dobivenih okosnicom naučenom SimCLR-om.