

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1282

**DETEKCIJA UMJETNO GENERIRANIH SLIKA PRIMJENOM
METODA STROJNOG I DUBOKOG UČENJA**

Josipa Sever

Zagreb, lipanj 2026.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1282

**DETEKCIJA UMJETNO GENERIRANIH SLIKA PRIMJENOM
METODA STROJNOG I DUBOKOG UČENJA**

Josipa Sever

Zagreb, lipanj 2026.

DIPLOMSKI ZADATAK br. 1282

Pristupnica: **Josipa Sever (0036538934)**

Studij: Računarstvo

Profil: Računarska znanost

Mentor: prof. dr. sc. Alan Jović

Zadatak: **Detekcija umjetno generiranih slika primjenom metoda strojnog i dubokog učenja**

Opis zadatka:

Razvoj generativnih modela umjetne inteligencije omogućio je stvaranje vizualno uvjerljivih sintetičkih slika koje je sve teže razlikovati od stvarnih. Takve tehnologije otvaraju brojne mogućnosti primjene, ali istodobno stvaraju izazove u području sigurnosti, digitalne forenzike, zaštite autentičnosti sadržaja i sprječavanja dezinformacija. Zbog toga detekcija umjetno generiranih slika predstavlja aktualan istraživački problem s rastućom znanstvenom i praktičnom važnošću. Zadatak rada je istražiti problem detekcije umjetno generiranih slika analizom postojećih pristupa iz literature. Pritom je potrebno eksperimentalno ispitati i usporediti različite metode strojnog i dubokog učenja koje se mogu upotrijebiti za automatizaciju detekcije (npr. GAN, difuzijski modeli). Potrebno je identificirati značajke i obrasce koji razlikuju stvarne i generirane slike, primijeniti odabrane pristupe strojnog i dubokog učenja, te provesti sustavno vrednovanje njihovih performansi na javno dostupnim skupovima podataka primjenom odgovarajućih metrika. Posebnu pozornost treba posvetiti analizi učinkovitosti metoda te sposobnosti modela da generalizira na skupove podataka koji nisu korišteni tijekom učenja.

Rok za predaju rada: 26. lipnja 2026.

Sadržaj

1. Uvod	3
2. Teorijske osnove	4
2.1. Duboko učenje	4
2.1.1. Generativni modeli	4
2.1.2. Diskriminativni modeli	9
2.2. Mjere vrednovanja	13
3. Postojeće metode detekcije	15
3.1. Detekcija na temelju artefakata	15
3.1.1. CNNDetection	16
3.1.2. PatchCraft	16
3.2. Detekcija na temelju slikovnih kodera	18
3.2.1. Universal Fake Detect (UFD)	19
4. Skup podataka	21
4.1. Skup za učenje	21
4.2. Skup za testiranje	22
5. Eksperimenti postojećih metoda	23
5.1. Rezultati	23
5.1.1. CNNDetection	23
5.1.2. PatchCraft	26
5.1.3. Universal Fake Detect	29
6. Adaptivna fuzija CNNDetection i Universal Fake Detect metode	33
6.1. Arhitektura	33

6.2. Implementacija	35
6.3. Rezultati	36
7. Zaključak	41
Literatura	42
Sažetak	46
Abstract	47

1. Uvod

S popularizacijom generativnih modela dubokog učenja značajno je porasla prisutnost umjetno generiranih vizualnih sadržaja. Takvi sadržaji postali su dio svakodnevnog života, a napretkom generativnih tehnika sve je teže razlikovati stvarne od sintetičkih slika. Iako se često koriste u zabavne svrhe, mogu imati i zlonamjerne primjene poput prevara, širenja dezinformacija ili lažnog predstavljanja. Nemogućnost pouzdanog prepoznavanja takvih sadržaja dodatno olakšava njihovu zlouporabu, zbog čega razvoj metoda za njihovu detekciju predstavlja važan istraživački izazov.

Razvijene su brojne metode detekcije umjetno generiranih slika, no sve dijele zajednički temeljni izazov: generalizaciju na neviđene generativne modele. Naučiti model prepoznavanja na svim mogućim generativnim arhitekturama gotovo je nemoguće jer novi modeli stalno nastaju, a skup podataka za učenje potencijalno bi bio beskonačno velik. Stoga je ključno razviti metode koje, naučene na što manjem broju generativnih modela, ipak zadržavaju sposobnost generalizacije na nove, neviđene izvore.

Ovaj rad analizira i uspoređuje tri postojeće metode detekcije generiranih slika: CNN-Detection, PatchCraft i Universal Fake Detect. Dodatno, predložena je i implementirana nova metoda koja kombinira dvije od njih, s ciljem poboljšanja generalizacije.

2. Teorijske osnove

2.1. Duboko učenje

Duboko učenje je grana strojnog učenja temeljena na višeslojnim neuronskim mrežama koje omogućuju automatsko učenje značajki iz podataka. Zbog sposobnosti samostalnog izdvajanja složenih i apstraktnih obrazaca iz velikih skupova podataka, duboko učenje široko se primjenjuje u računalnom vidu, području koje obuhvaća i problem detekcije umjetno generiranih slika. U kontekstu ovog rada, modeli dubokog učenja mogu se podijeliti na generativne i diskriminativne. Pritom prvi služe za generiranje novih podataka, dok drugi služe za klasifikaciju postojećih.

2.1.1. Generativni modeli

Za uspješnu detekciju umjetno generiranih slika, najprije je potrebno razumjeti kako generativni modeli rade. Povijest umjetnih generatora slika doseže do 1970-ih, kada je britanski slikar Harold Cohen razvio računalni sustav nazvan AARON. Taj je sustav u početku generirao jednostavne crteže, a kasnije i obojene slike. Sustav se, za razliku od modernih generativnih modela, temeljio na unaprijed definiranim pravilima [1]. Iako su takvi pristupi predstavljali važan korak u računalnom generiranju slika, značajniji napredak ostvaren je tek pojavom modela dubokog učenja, koji su omogućili generiranje fotorealističnih i kompleksnijih sadržaja. Duboki generativni modeli uče distribuciju podataka iz skupa za učenje kako bi generirali vjerodostojne podatke iz te distribucije. Među najznačajnijim pristupima generiranju slika ističu se generativne suparničke mreže, difuzijski modeli i transformerske arhitekture, detaljnije opisani u nastavku. Uz navedene arhitekture u radu se koristi i Midjourney v6, generativni model čija arhitektura nije javno objavljena[2].

Generativne suparničke mreže (GAN)

Generativne suparničke mreže, odnosno GAN-ovi, uvedene su 2014. godine. Još su nekoliko godina nakon toga ostale među najuspješnijim generativnim modelima.

GAN-ovi se sastoje od dviju dubokih neuronskih mreža: generatora i diskriminatora. Te se dvije mreže tijekom učenja natječu jedna protiv druge. Generator na temelju slučajnog šuma stvara sintetičke slike, nastojeći pritom da budu što sličnije stvarnim slikama. Zadatak diskriminatora je razlikovanje stvarnih slika od generiranih. Učenje napreduje sve dok generator ne postane dovoljno dobar da diskriminator više ne može pouzdano razlikovati stvarne i generirane slike.

Ciljna funkcija dana je izrazom (2.1). Generator ju minimizira, a diskriminator maksimizira. U izrazu x označava stvarne slike, z slučajni šum, $D(x)$ vjerojatnost da je slika x stvarna, a $G(z)$ generiranu sliku [3].

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))] \quad (2.1)$$

GAN-ovi se mogu podijeliti na bezuvjetne i uvjetne. Bezuvjetni generiraju slike iz slučajnog šuma bez dodatnih uvjeta, kao što je prethodno opisano. Uvjetni GAN-ovi proširuju osnovnu arhitekturu dodatnom informacijom, poput oznake klase ili ulazne slike, koja usmjerava proces generiranja[4]. U ovom radu koristit će se slike generirane objema vrstama GAN-ova.

Bezuovjetni GAN-ovi korišteni u ovom radu su ProGAN, BigGAN, StyleGAN i StyleGAN2. Učenje ProGAN-a započinje slikom niske rezolucije te plitkim generatorom i diskriminatorom. Rezolucija slike se postupno povećava dodavanjem slojeva u mreže, što osigurava vjerodostojnije slike visoke rezolucije i brže učenje[5]. Arhitektura BigGAN-a je, s druge strane, fiksna, ali je veći model od ProGAN-a s dva do četiri puta više parametara i osam puta većim podskupom primjeraka za učenje (engl. *batch*). Povećanjem modela postignuta je viša rezolucija i kvaliteta generiranih slika. Uz to, uvedene su i određene modifikacije regularizacije kako bi se očuvala stabilnost učenja[6]. Nadalje, generator StyleGAN-a modificiran je tako što umjesto nasumičnog šuma koristi zasebnu mrežu za mapiranje u stilski međuprostor. Time je omogućena kontrola stilskih obilježja

generiranih slika[7]. Modelom StyleGAN2 uvedene su dodatne promjene u arhitekturi i procesu učenja kako bi se spriječili artefakti StyleGAN-a te unaprijedila kvaliteta slike[8].

Uvjetni GAN-ovi korišteni u ovom radu su CycleGAN, GauGAN i StarGAN. CycleGAN se primjenjuje za *image-to-image* translaciju, poput prijenosa stila i poboljšanja slike. Od ostalih metoda za taj problem razlikuje se po tome što ne uči izravno na parovima slika, već na neuparenim podacima pomoću gubitka cikličke konzistentnosti koji osigurava da je translacija ciklički reverzibilna[9]. StarGAN se također primjenjuje za *image-to-image* translaciju, no za razliku od CycleGAN-a, koji je ograničen samo na dvije domene, omogućuje translaciju između više različitih domena. Kako bi to postigao dodatno optimira gubitak klasifikacije domene s pomoćnim klasifikatorom ponad diskriminatora[10]. GauGAN uvodi prostorno prilagodljivu normalizaciju koja rješava problem gubitka semantičke informacije u normalizacijskim slojevima. Takva normalizacija koristi semantičke mape, odnosno segmentacijske maske, koje čuvaju bitne prostorne informacije kroz mrežu. Time je omogućeno generiranje dosljednih fotorealističnih prikaza[11].

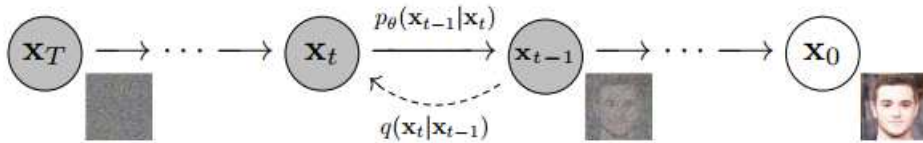
Difuzijski modeli

Difuzijski modeli stvaraju slike postupnim uklanjanjem šuma, krenuvši od nasumičnog Gaussovog šuma. Difuzijski model definira unaprijedni proces dodavanja šuma i unazadni proces uklanjanja šuma. Drugi, unazadni prolaz, kod prvih se difuzijskih modela provodio u prostoru piksela, no kod modernijih, poput onih korištenih u ovom radu, provodi se u latentnom prostoru zbog smanjenja računalne složenosti. U nastavku je opisana izvorna arhitektura difuzijskog modela.

Tijekom unaprijednog prolaza model slici dodaje nasumičan šum kroz mnogo vremenskih koraka, čime slika postaje gotovo isključivo Gaussov šum. Zatim u unazadnom prolazu model uči korak po korak predvidjeti i ukloniti šum kako bi rekonstruirao originalnu sliku. Oba se procesa modeliraju kao Markovljevi lanci, što znači da svaki vremenski korak ovisi isključivo o prethodnom. Učenje se formalizira funkcijom cilja koja minimizira razliku između dodanog šuma i šuma kojeg model predviđa u unazadnom prolazu. Funkcija cilja dana je izrazom 2.2, gdje ϵ označava stvarno dodani šum, $\epsilon_{\theta}(x_t, t)$ šum koji model predviđa za zašumljenu sliku x_t u trenutku t , a θ parametre modela.

Nakon što je naučen, model generira sliku iterativnim provođenjem unazadnog procesa kroz niz vremenskih koraka. Model s naučenim fiksnim parametrima kao ulaz prima nasumični šum pa ga iterativno uklanja do realistične slike, što je prikazano na slici 2.1.

$$L_{simple} = \mathbb{E}_{t, x_0, \epsilon} [\|\epsilon - \epsilon_{\theta}(x_t, t)\|^2] \quad (2.2)$$



Slika 2.1. Grafički prikaz difuzijskog modela. Strelica ulijevo prikazuju unaprijedni prolaz dodavanja šuma, a strelice udesno unazadni prolaz uklanjanja šuma.[12]

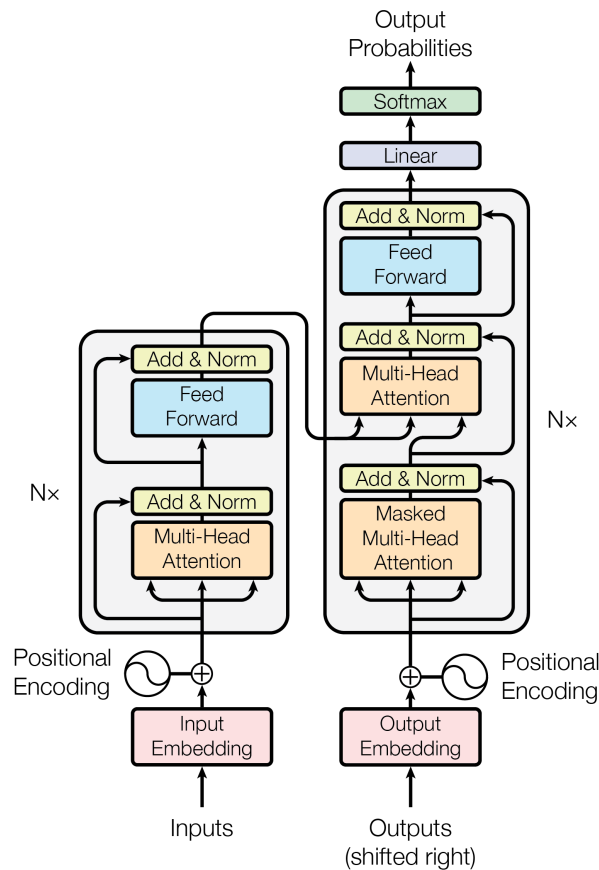
U ovom radu korišteni su modeli iz skupine latentnih difuzijskih modela (LDM). Ti modeli cjelokupni proces dodavanja i uklanjanja šuma umjesto u prostoru piksela obavljaju u kompaktnijem, latentnom prostoru dobivenom pomoću autoenkodera. Konkretno, korištene su novije varijante arhitekture Stable Diffusion. Originalni Stable Diffusion model koristi CLIP ViT-L/14 kao zamrznuti tekstni koder[13]. Kasniji Stable Diffusion 2.1 prelazi na OpenCLIP-ViT/H[13]. Stable Diffusion XL predstavlja velik skok u kvaliteti slike, koristi oba prethodno spomenuta tekstna koder i tri puta više parametara u okosnici[14]. Stable Diffusion 3 prelazi na arhitekturu s transformerom kao okosnicom, koristi Google T5-XXL kao treći tekstni koder, što omogućava precizno ispisivanje čitljivog teksta unutar slike i razumijevanje kompleksnih prostornih odnosa te uvodi novu matematičku formulaciju rasporeda difuzije. Stable Diffusion 3.5 je poboljšana verzija prethodnog modela sa stabilnijim učenjem i poboljšanom kvalitetom slike[15].

Transformerske arhitekture

Transformeri su modeli dubokog učenja temeljeni na mehanizmu samopažnje, izvorno razvijeni za obradu sekvencijalnih podataka poput teksta. Temelje se na arhitekturi koder-dekoder u kojoj koder iz ulaznih podataka izvlači reprezentacije, a dekoder na temelju njih generira izlaz. Iako su izvorno namijenjeni obradi teksta, transformeri su se pokazali uspješnima i u području računalnog vida, gdje se koriste za klasifikaciju i

generiranje slika.

Arhitektura transformera prikazana je na slici 2.2. Sastoji se od koder i dekodera koji oboje koriste mehanizme pažnje. Koder obrađuje ulazne sekvence u informativne visokodimenzijske reprezentacije, na temelju kojih dekoder generira izlaz.



Slika 2.2. Arhitektura transformera: lijevi dio je koder, a desni dekoder[16]

Koder se sastoji od šest slojeva čiji su podslojevi mehanizam samopažnje s više glava (engl. *Multi-Head Self-Attention*) i unaprijedna neuronska mreža (engl. *Feed Forward Neural Network*). Samopažnja omogućuje koderu određivanje važnosti pojedinih dijelova ulazne sekvence neovisno o njihovoj poziciji, a unaprijedna mreža zatim obrađuje dobivene reprezentacije.

Dekoder se također sastoji od šest slojeva, no dodatno sadrži podsloj pažnje koder-dekoder. Mehanizam samopažnje s više glava maskiran je kako dekoder ne bi imao pristup još negeneriranim dijelovima izlazne sekvence. Pažnja koder-dekoder usmjerava dekoder na relevantne dijelove koderskih reprezentacija, a unaprijedna mreža obrađuje kombinirani izlaz oba mehanizma pažnje[16].

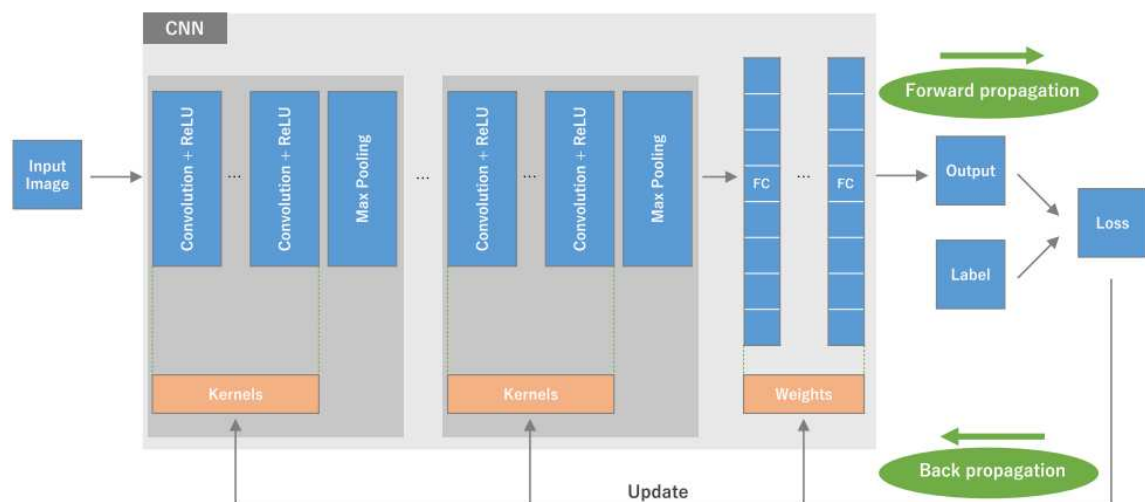
U ovom se radu kao primjer generativnog transformerskog modela koristi DALL-E 3, multimodalni generativni model tvrtke OpenAI koji generira slike na temelju tekstnog opisa. U odnosu na prethodne verzije, DALL-E 3 bolje razumije i slijedi tekstne upute, što rezultira kvalitetnijim generiranim slikama. Arhitekturno se temelji na difuzijskom modelu s transformerom kao okosnicom [17].

2.1.2. Diskriminativni modeli

Diskriminativni modeli dubokog učenja prikladni su za detekciju umjetno generiranih slika budući da se taj problem može formulisati kao binarna klasifikacija slika u stvarne i generirane. U ovom se radu konkretno koriste modeli rezidualnih neuronskih mreža i model temeljen na transformerskoj arhitekturi, opisani u nastavku.

Konvolucijske neuronske mreže (CNN)

Konvolucijske neuronske mreže vrsta su dubokih neuronskih mreža posebno prilagođenih obradi slikovnih podataka. Posebne su po tome što omogućuju hijerarhijsko učenje značajki i prostornih obrazaca različite kompleksnosti te detekciju objekata neovisno o njihovoj poziciji u slici. Kako bi to postigle, koriste tri vrste slojeva: konvolucijske, potpuno povezane i slojeve sažimanja. Konvolucijska neuronska mreža obično se sastoji od više naslaganih blokova tih slojeva, kao što je prikazano na slici 2.3.

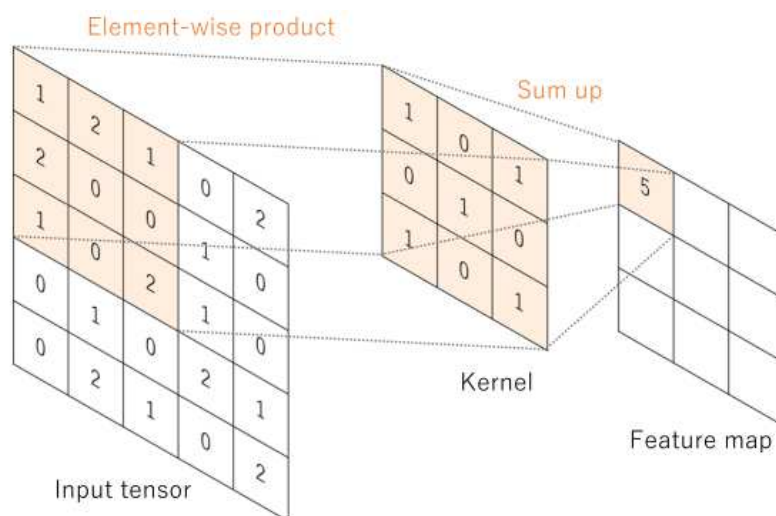


Slika 2.3. Arhitektura konvolucijske neuronske mreže s konvolucijskim blokovima, slojevima sažimanja i potpuno povezanim slojevima.[18]

Konvolucijski blokovi sastoje se od nekoliko konvolucijskih slojeva s ReLU aktivacij-

skom funkcijom, nakon kojih slijedi sloj sažimanja maksimumom. Konvolucija se provodi odgovarajućom jezgrom. Nakon više takvih blokova slijedi jedan ili više potpuno povezanih slojeva. U procesu učenja, u unaprijednom prolazu računa se izlaz i gubitak mreže, dok se u unazadnom prolazu ažuriraju parametri mreže: konvolucijske jezgre i težine potpuno povezanih slojeva.

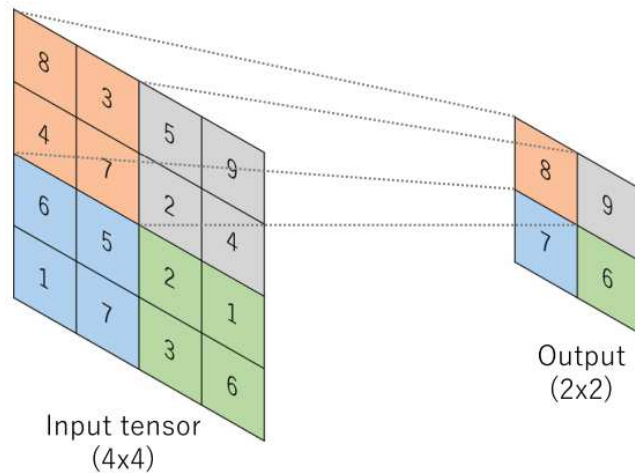
Osnovna komponenta konvolucijskih neuronskih mreža su konvolucijski slojevi koji provode ekstrakciju značajki. U tim slojevima provodi se matematička operacija konvolucije: jezgra zadanih dimenzija klizi po ulaznoj slici s određenim korakom (engl. *stride*), najčešće jednakim 1. Na svakoj poziciji računa se skalarni produkt jezgre i odgovarajućeg područja ulazne slike, čime se dobiva jedna vrijednost u izlaznoj mapi značajki, kao što je prikazano na slici 2.4. Kako bi se kontrolirale dimenzije izlazne mape značajki i očuvale informacije na rubovima slike, ulazna se slika može proširiti dodatnim vrijednostima oko rubova. Taj postupak naziva se nadopunjavanje (engl. *padding*). Najčešće se koristi nadopunjavanje nulama (engl. *zero padding*), pri čemu odabir veličine nadopunjavanja određuje hoće li dimenzije izlazne mape biti jednake ulaznoj ili manje.



Slika 2.4. Primjer konvolucije s jezgrom veličine 3×3 , bez nadopune, s korakom 1[18]

Slojevi sažimanja smanjuju prostorne dimenzije mape značajki, čime se reducira broj parametara i računalna složenost mreže te postiže određena invarijantnost na translacije. Poput konvolucijskog sloja, i sloj sažimanja koristi jezgru koja klizi po ulaznoj mapi značajki, no korak je najčešće jednak veličini jezgre kako se područja ne bi preklapala. Kod sažimanja maksimumom na svakoj poziciji jezgre u izlaz se upisuje maksimalna vrijednost iz odgovarajućeg područja ulazne mape, kao što je prikazano na slici 2.5. Osim saži-

manja maksimumom, ponekad se koristi sažimanje prosječnom vrijednošću, pri kojem se na svakoj poziciji jezgre u izlaz upisuje prosjek vrijednosti odgovarajućeg područja.



Slika 2.5. Primjer sažimanja maksimumom s jezgrom veličine 2×2 i korakom 2[18]

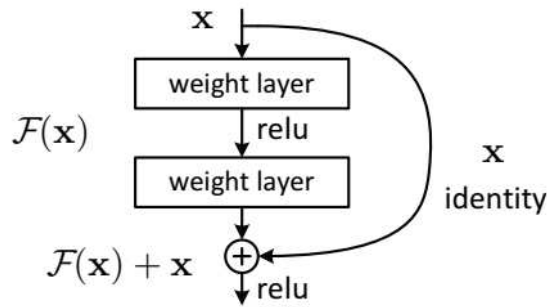
Zadnji, potpuno povezani slojevi kombiniraju izlučene značajke kako bi proizveli konačni izlaz mreže. Za razliku od konvolucijskih slojeva, svaki neuron u potpuno povezanom sloju povezan je sa svim neuronima prethodnog sloja, pri čemu svaka veza ima odgovarajuću naučenu težinu. Prije potpuno povezanog sloja mapa značajki se splošnjava u jednodimenzionalni vektor. U zadnjem potpuno povezanom sloju primjenjuje se aktivacijska funkcija prikladna za zadatak. Za binarnu klasifikaciju to je sigmoidna funkcija, koja izlaz preslikava u raspon $[0, 1]$ i interpretira se kao vjerojatnost pripadnosti pojedinoj klasi[18].

Rezidualne neuronske mreže

Rezidualna neuronska mreža vrsta je konvolucijske mreže oblikovane za efikasno učenje vrlo dubokih arhitektura. Tradicionalne duboke mreže suočavaju se s problemom nestajućeg ili eksplodirajućeg gradijenta, koji učenje čini nestabilnim, te problemom degradacije. To je pojava u kojoj pogreška učenja prestaje padati, a ponekad i raste, nakon što dubina mreže prijeđe određeni prag. Rezidualne neuronske mreže rješavaju oba problema uvođenjem rezidualne veze koja stabilizira i optimira proces učenja.

Rezidualne veze (ili skip-veze) preskaču jedan ili više slojeva u mreži, kao što je prikazano na slici 2.6. Definirane su kao identitetsko preslikavanje koje ulaz prosljeđuje izravno i zbraja ga s izlazom preskočenih slojeva. Na slici $\mathcal{F}(\mathbf{x})$ označava izlaz preskoče-

nih slojeva, a $\mathcal{F}(\mathbf{x}) + \mathbf{x}$ konačni izlaz bloka nakon zbrajanja. Budući da rezidualne veze ne uvode dodatne parametre, računalna složenost mreže ostaje nepromijenjena. Ključna prednost takvih veza jest mogućnost izravnog toka gradijenta kroz mrežu tijekom unazadnog prolaza, čime se ublažava problem nestajućeg gradijenta i omogućuje stabilno učenje vrlo dubokih arhitektura.



Slika 2.6. Blok s rezidualnom vezom[19]

U ovom radu koristi se rezidualna mreža ResNet50. Ta široko korištena rezidualna mreža sadrži 50 slojeva koja uvođenjem blokova uskog grla (engl. *bottleneck*) omogućuje znatno povećanje dubine mreže. Svaki takav blok sastoji se od tri konvolucijska sloja: prvi smanjuje broj kanala 1×1 konvolucijom, srednji obrađuje značajke 3×3 konvolucijom, a zadnji nazad povećava broj kanala 1×1 konvolucijom. Obje mreže inicijaliziraju se težinama prethodno naučenim na skupu ImageNet[19].

Transformerske arhitekture

Prethodno opisane transformerske arhitekture mogu se koristiti i u diskriminativnom smislu tako da se naučene reprezentacije koriste za razlikovanje klasa. U ovom se radu tako koristi model CLIP ViT-L/14. CLIP (engl. *Contrastive Language-Image Pretraining*) je model tvrtke OpenAI koji uči zajedničke reprezentacije slika i teksta kontrastivnim učenjem na velikom skupu parova slika i tekstnih opisa prikupljenih s interneta [20]. Sastoji se od tekstnog i vizualnog koda koji ulaze preslikavaju u zajednički prostor reprezentacija. Kao vizualni koder koristi se Vision Transformer (ViT) [21], koji ulaznu sliku dijeli na zakrpe fiksne veličine i obrađuje ih kao sekvencu transformerom. Konkretna varijanta korištena u ovom radu je ViT-L/14, gdje L označava veličinu modela, a 14 veličinu zakrpe u pikselima. Budući da je CLIP učen na raznolikim podacima, naučene reprezentacije dobro generaliziraju na slike iz različitih domena, što ga čini pogodnim

za detekciju generiranih slika neovisno o generativnom modelu.

2.2. Mjere vrednovanja

Sposobnost generalizacije pojedinih metoda detekcije procjenjuje se mjerama vrednovanja, koje služe i za međusobnu usporedbu metoda. Kao osnovna mjera koristi se točnost, no budući da sama po sebi može biti nedovoljna pri neuravnoteženim skupovima podataka, u ovom se radu dodatno koristi mjera F1. Za detaljniji uvid u ponašanje klasifikatora prikazuje se i matrica zabune, iz koje se mogu iščitati vrste pogrešaka koje metoda čini.

Klasifikacija slika u stvarne i generirane binarna je klasifikacija, gdje 0 označava stvarnu, a 1 generiranu sliku, pa su pri označavanju slike moguća četiri slučaja:

- stvarni pozitiv (engl. *true positive*, *TP*): slika je generirana i označena je kao generirana
- lažni pozitiv (engl. *false positive*, *FP*): slika je stvarna, a označena je kao generirana
- lažni negativ (engl. *false negative*, *FN*): slika je generirana, a označena je kao stvarna
- stvarni negativ (engl. *true negative*, *TN*): slika je stvarna i označena je kao stvarna

S tako definiranim slučajevima matrica zabune prikazuje se prema tablici 2.1., gdje TP označava broj stvarnih pozitiva, FP broj lažnih pozitiva, FN broj lažnih negativa, a TN broj stvarnih negativa.

Tablica 2.1. Matrica zabune za binarnu klasifikaciju

		Stvarna oznaka	
		1	0
Predviđena oznaka	1	TP	FP
	0	FN	TN

Točnost (engl. *accuracy*) se definira kao udio točno klasificiranih primjera u skupu svih primjera, a računa se prema izrazu 2.3. Ta mjera je intuitivna, no nije uvijek prikladna jer u slučaju neuravnoteženih skupova podataka može biti zavaravajuća. Primjerice, ako skup za testiranje sadrži znatno više stvarnih nego generiranih slika, klasifikator

koji sve slike označi kao stvarne postiže visoku točnost, ali zapravo nije uspješan. Zato se uvode dodatne mjere vrednovanja koje su informativnije u tom slučaju.

$$Acc = \frac{TP + TN}{TP + FP + FN + TN} \quad (2.3)$$

Preciznost (engl. *precision*) je udio stvarno pozitivnih primjera u skupu svih primjera koje je klasifikator označio pozitivno. Računa se prema sljedećem izrazu:

$$P = \frac{TP}{TP + FP} \quad (2.4)$$

Odziv (engl. *recall*) je udio stvarno pozitivnih primjera u svim pozitivnim primjerima. Računa se prema sljedećem izrazu:

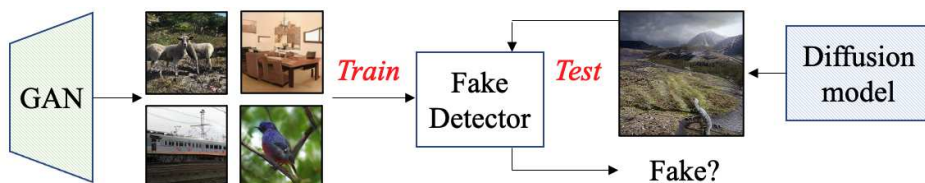
$$R = \frac{TP}{TP + FN} \quad (2.5)$$

Pomoću preciznosti i odziva definira se mjera F1 kao njihova harmonijska sredina. Računa se prema izrazu 2.6 Preciznost, odziv i mjera F1 poprimaju vrijednost iz intervala $[0, 1]$, pri čemu 1 označava savršenu klasifikaciju. Mjera F1 postiže visoku vrijednost samo ako su obje njene komponente, preciznost i odziv, istovremeno visoke, što ju čini robusnijom mjerom od točnosti u slučaju neuravnoteženih skupova podataka.

$$F1 = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{P + R} \quad (2.6)$$

3. Postojeće metode detekcije

Metode detekcije umjetno generiranih slika temelje se na klasifikatorima koji slike svrstavaju u dvije kategorije: generirane i stvarne. S obzirom na to da ne postoji samo jedna vrsta generatora te da su generativni modeli sve brojniji i uspješniji, metode detekcije bi trebale biti primjenjive na slike bilo kojeg generativnog modela. Međutim, svim je metodama zajednički problem prenaučivosti klasifikatora na generativni model korišten u procesu učenja. Iako su metode uspješne na slikama istog modela, često se javlja problem generalizacije na slike neviđenih modela. Stoga svaka od razmatranih metoda uvodi poseban mehanizam poboljšanja generalizacije s ciljem razvoja univerzalnog detektora, prikazanog na slici 3.1.



Slika 3.1. Univerzalni detektor treba biti primjenjiv na slike bilo kojeg modela koji nije korišten tijekom učenja. Prikazani detektor učen na slikama generiranim GAN-om treba prepoznavati slike generirane difuzijskim modelom.[22]

Metode detekcije umjetno generiranih slika mogu se podijeliti u kategorije prema vrsti značajki koje koriste za razlikovanje stvarnih slika od sintetičkih. Ovaj rad istražuje metode detekcije na temelju artefakata i na temelju slikovnih kodera, prema podjeli iz rada [23].

3.1. Detekcija na temelju artefakata

Metode detekcije na temelju artefakata polaze od pretpostavke da generativni modeli u slikama ostavljaju prepoznatljive lokalne nepravilnosti, ponavljajuće obrasce ili frekven- cijske tragove. Ove metode ne oslanjaju se na semantički sadržaj slike, već na vizualne

anomalije koje su ljudskom oku često neprimjetne, no dovoljno su konzistentne da ih klasifikator može naučiti.

U ovom radu analizirana je metoda CNNDetection kao klasični primjer detekcije generiranih slika na temelju artefakata i metoda PatchCraft koja naglašava teksturne zakrpe i odnose među pikselima kao izvor informativnih artefakata.

3.1.1. CNNDetection

Metoda CNNDetection[24] pristup je detekciji s ciljem stvaranja univerzalnog detektora umjetno generiranih slika koji dobro generalizira na svim CNN generatorima. Rad pokazuje da konvolucijska neuronska mreža učena isključivo na ProGAN slikama postiže dobru generalizaciju na druge CNN modele. Metoda koristi ResNet-50 arhitekturu, prethodno učenu na ImageNetu, a zatim fino podešenu kao binarni klasifikator za razlikovanje stvarnih i umjetno generiranih slika.

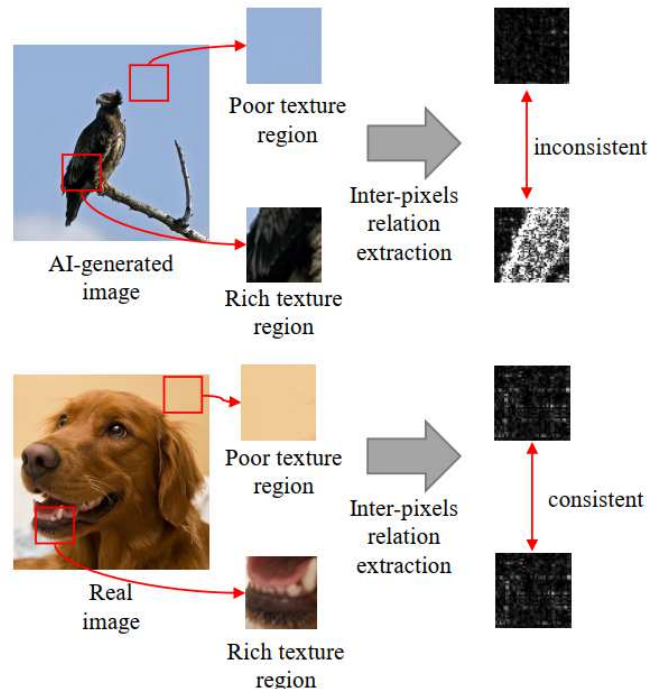
Augmentacija i raznolikost podataka imaju ključnu ulogu u ovoj metodi detekcije. Podaci su augmentirani tako što je s određenom vjerojatnošću na ulazne slike primijenjeno Gaussovo zamućenje i JPEG kompresija. Time je povećana generalizacija detektora. Budući da slike dijeljenjem na Internetu prolaze kroz različite metode obrade poput kompresije i filtriranja, augmentacijom se model priprema na takve uvjete, čime je također povećana i robusnost detektora u stvarnom okruženju.

Ograničenje metode je što je osmišljena primarno za detekciju slika generiranih GAN-ovima. Kasnija istraživanja pokazuju da model slabije generalizira na slikama generiranim difuzijskim modelima i transformerima, koji unose drugačije vrste artefakata od onih karakterističnih za arhitekture GAN-ova[23].

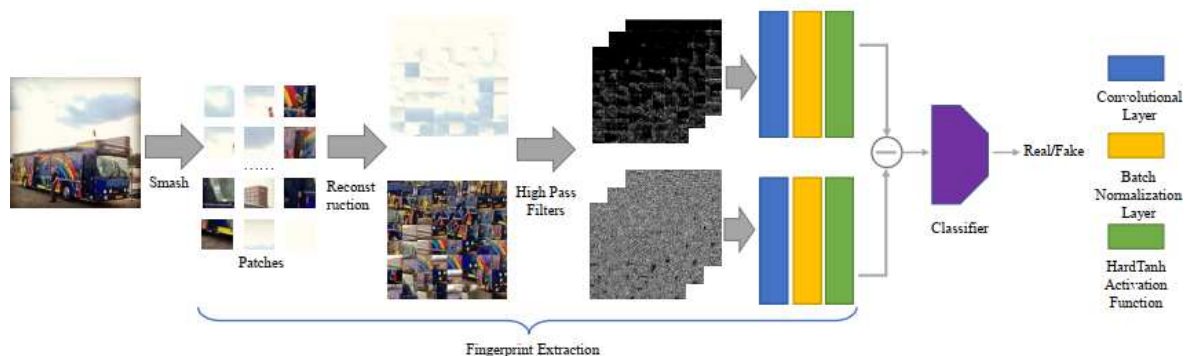
3.1.2. PatchCraft

PatchCraft je metoda detekcije umjetno generiranih slika koja se manje oslanja na globalne semantičke značajke. Umjesto toga koristi teksturne zakrpe (engl. *patches*) i međupikselne korelacije. Motivacija za takav pristup prikazana je na slici 3.2., a proizlazi iz činjenice da su međupikselne korelacije između teksturno bogatih i siromašnih regija u stvarnim slikama konzistentne, dok u generiranim slikama nisu. U tu svrhu uvodi

postupak *Smash&Reconstruction* i predlaže univerzalni otisak prsta prisutan u generiranim slikama. Cjelokupni postupak detekcije umjetno generiranih slika ovom metodom prikazan je na slici 3.3. i opisan u nastavku.



Slika 3.2. Motivacija za uvođenje međupikselnih korelacija između regija različitih tekstura: kod stvarnih slika su konzistentne, dok kod generiranih nisu.[25]



Slika 3.3. Detekcija umjetno generiranih slika PatchCraft metodom[25]

Smash&Reconstruction je postupak predobrade slike koji uništava semantičku informaciju slike, dok istovremeno naglašava teksturne značajke. U prvoj, odnosno *Smash* fazi, ulazna slika nasumično se reže na zakrpe veličine 32×32 piksela. Zatim se za svaku zakrpu mjeri raznolikost teksture tako što se sumiraju razlike između vrijednosti susjednih piksela u četiri smjera: dijagonalno, kontra-dijagonalno, horizontalno i vertikalno. Zakrpe bogate teksture imaju veliku raznolikost, dok one siromašne teksture imaju ni-

sku. U *Reconstruction* fazi zakrpe se sortiraju po raznolikosti i grade se dvije nove slike. Jedna je slika bogate teksture koja se sastoji od zakrpa s gornjih 33% raznolikosti, a druga je slika siromašne teksture s donjih 33% raznolikosti.

Idući ključni korak je ekstrakcija univerzalnog otiska prsta. Temelji se na analizi visokofrekventnih komponenti i kontrastu između regija različitih tekstura. Dvije rekonstruirane slike prolaze kroz skup od 30 različitih visokopropusnih filtara koji dodatno potiskuju semantičku informaciju, a naglašavaju međupikselnu korelaciju. Svaka od filtriranih slika zatim ulazi u učljivi konvolucijski blok koji izvlači relevantne značajke međupikselne korelacije filtriranih slika. Univerzalni otisak prsta označava međupikselnu korelaciju kontrasta između regija bogate i siromašne teksture, a dobiven je kao rezidual izlaza konvolucijskog bloka.

Na kraju se ekstrahirani otisak prsta prosljeđuje u binarni klasifikator koji ulaznu sliku označuje kao stvarnu ili generiranu. Klasifikator se sastoji od četiri kaskadna konvolucijska bloka. Takva je struktura dovoljna za detekciju sintetičkih slika jer je razlika između univerzalnog otiska prsta takvih i stvarnih slika velika. Tijekom učenja cjelokupnog modela minimizira se gubitak unakrsne entropije, pri čemu se ažuriraju parametri klasifikatora i konvolucijskih blokova u modulu za ekstrakciju otiska prsta[25].

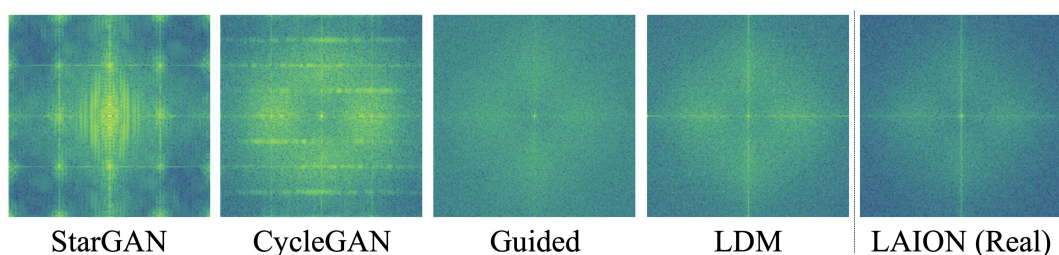
3.2. Detekcija na temelju slikovnih kodera

Metode detekcije na temelju slikovnih kodera polaze od pretpostavke da prethodno naučeni vizualni modeli, poput CLIP-a, mogu izdvajati reprezentacije slike koje su dovoljno informativne da razlikuju stvarne slike od generiranih. Za razliku od metoda temeljenih na artefaktima koje traže artefakte u pikselnom ili frekvencijskom prostoru, ove metode koriste semantički i višerazinski prostor značajki slikovnog kodera za klasifikaciju slike kao stvarne ili generirane.

U ovom radu analizirana je metoda Universal Fake Detect (UFD)[22] kao primjer detekcije na temelju slikovnih kodera, opisana u nastavku.

3.2.1. Universal Fake Detect (UFD)

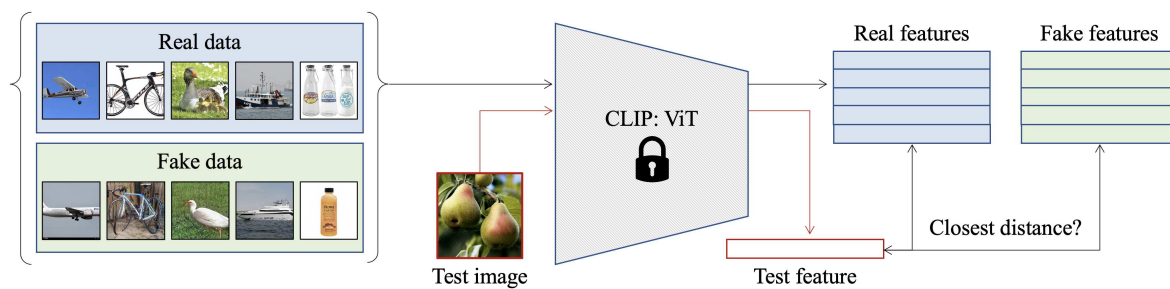
Metoda Universal Fake Detect predlaže detekciju generiranih slika bez eksplicitnog učenja artefakata. Metode temeljene na artefaktima uspješno prepoznaju slike generirane modelima sličnim onome na kojemu je klasifikator učen, odnosno slike koje sadrže dovoljno slične artefakte. Međutim, slike koje sadrže artefakte neviđenih modela takve metode često pogrešno svrstavaju u stvarne, budući da ti artefakti nisu dovoljno slični onima viđenima u skupu za učenje (Sl. 3.4.). Stoga se ova metoda ne oslanja na učenje značajki koje razlikuju generirane slike od stvarnih, već koristi fiksni CLIP-ViT prostor značajki.



Slika 3.4. Prosječni frekvencijski spektar slika generiranih različitim modelima i stvarnih slika. StyleGAN i CycleGAN pripadaju GAN-ovima, dok su Guided i LDM difuzijski modeli. Vidljivo je da svaka familija ostavlja karakteristične obrasce u frekvencijskoj domeni koji se međusobno razlikuju.[22]

Budući da cilj metode nije učenje značajki karakterističnih za generirane slike, nije potrebno učiti model za ekstrakciju značajki. Umjesto toga dovoljno je preslikavanje piksela u neki prostor značajki jer pikseli sami nisu dovoljno informativni za detekciju. Taj prostor mora biti naučen na velikom i raznovrsnom skupu podataka te moći uhvatiti detalje kako bi bio dovoljno ekspresivan za reprezentaciju slika iz različitih domena. Zato je za okosnicu odabran prethodno naučeni CLIP:ViT-L/14, odnosno njegov koder. Važno je naglasiti da taj koder nije učen za zadatak detekcije generiranih slika. Kroz koder su propuštene stvarne i generirane slike, čime su izgrađena dva odvojena skupa značajki: skup značajki stvarnih slika i skup značajki generiranih slika, što je prikazano na slici 3.5.

Nakon što su izgrađeni skupovi značajki, slika se klasificira kao stvarna ili generirana jednom od sljedećih dviju metoda: klasifikacijom najbližim susjednom ili linearnom klasifikacijom. U oba slučaja ulazna se slika najprije propušta kroz koder, čime su dobivene njene značajke koje se uspoređuju s prethodno spomenutim izgrađenim sku-



Slika 3.5. Detekcija generiranih slika UFD metodom pomoću najbližeg susjeda[22]

povima. Kod klasifikacije najbližim susjedom, za svaki skup značajki pronalazi se susjed najsličniji ulaznoj slici prema odabranoj mjeri udaljenosti. Ako je udaljenost do najbližeg susjeda iz skupa značajki stvarnih slika manja, ulaz se klasificira kao stvarna slika. U suprotnom je detektirana generirana slika. Kod linearne klasifikacije, konkretno logističke regresije, prethodno naučenom koderu dodan je jedan linearni sloj sa sigmoidnom aktivacijom. Taj sloj se uči za binarnu klasifikaciju pomoću gubitka binarne unakrsne entropije[22].

4. Skup podataka

4.1. Skup za učenje

Modeli iz postojećih metoda detekcije evaluirani su naučeni na skupovima podataka iz izvornih radova. Iznimka je metoda Universal Fake Detect koja je dodatno učena na novom skupu podataka. Za učenje nove predložene metode također je izgrađen novi skup podataka.

Skup za učenje metode CNNDetection sastoji se od slika generiranih ProGAN-om i stvarnih slika iz skupa LSUN[26], koji obuhvaća 20 kategorija različitih objekata. Za svaku kategoriju izgrađen je uravnotežen podskup od 36 000 slika. Time je dobiven skup od 720 000 slika, pri čemu je 360 000 stvarnih i 360 000 generiranih. Uz to je korišten i skup za validaciju koji se sastoji od ukupno 4 000 slika, 2 000 stvarnih i 2 000 generiranih. Autori metoda PatchCraft i Universal Fake Detect također za učenje su koristili isti skup podataka kao CNNDetection.

Za potrebe dodatnih eksperimenata s Universal Fake Detect metodom izgrađen je dodatni skup za učenje sa slikama generiranim LDM-om i stvarnim iz ImageNet-a[27]. Taj skup sastoji se od 20 000 slika generiranih LDM-om preuzetih iz rada [28] i 20 000 stvarnih slika, ukupno 40 000 slika. Dodatno je izgrađen skup za validaciju od ukupno 10 000 slika, 5 000 stvarnih i 5 000 generiranih. Na isti način, s jednakim omjerom i veličinom, izgrađen je i skup podataka za učenje predloženog modela s dvije grane.

Važno je napomenuti da su u radu [28] za učenje korištene stvarne slike iz skupa LSUN i COCO[29]. U ovom je radu unatoč tome odabran ImageNet jer bi u suprotnom došlo do preklapanja podataka u skupovima za učenje i testiranje pa ne bi bilo moguće ispravno procijeniti sposobnost generalizacije modela. Stoga je moguća slabija sposobnost generalizacije naučenog modela s dvije grane zbog nepodudarnosti distribucija stvarnih

i generiranih slika u skupu za učenje.

4.2. Skup za testiranje

Kako bi se procijenila sposobnost generalizacije razmatranih metoda izgrađen je skup za testiranje koji se sastoji od slika generiranih modelima iz različitih familija i odgovarajućih stvarnih slika. S obzirom na to da su generativni modeli sve uspješniji, kako bi eksperimenti bili relevantni, korištene su novije verzije modela. U tablici 4.1. prikazani su modeli korišteni za testiranje, uz veličinu odgovarajućih skupova. Za svaki generativni model skupu je pridružen jednak broj odgovarajućih stvarnih slika.

Tablica 4.1. Veličina pojedinih podskupova podataka generiranih slika u skupu za testiranje

Vrsta	Model	Veličina skupa
GAN	ProGAN	4 000
	BigGAN	2 000
	CycleGAN	1 321
	GauGAN	5 000
	StarGAN	1 999
	StyleGan	5 991
	StyleGAN2	7 988
Difuzijski	SD-2.1	10 017
	SD-3	10 017
	SD-3.5	10 017
	SDXL	10 017
Transformer	DALL-E3	10 017
Nepoznato	Midjourney	10 017

Slike generirane GAN-ovima preuzete su iz skupa za testiranje originalnog rada CN-NDetection metode. Taj dio skupa obuhvaća sljedeće modele: ProGAN, BigGAN, CycleGAN, GauGAN, StarGAN, StyleGAN i StyleGAN2. Za svaki od njih u skupu su i odgovarajuće stvarne slike iz skupa na kojem je pojedini model učen, što obuhvaća LSUN, COCO, ImageNet i CelebA[30].

Za difuzijske, transformerske i nepoznate arhitekture preuzet je skup COCO AI dostupan na platformi HuggingFace[31]. Taj skup sadrži slike generirane modelima StableDiffusion 2.1, StableDiffusion 3, StableDiffusion 3.5, StableDiffusion XL, DALL-E3 i Midjourney v6, uz odgovarajuće stvarne slike iz skupa COCO.

5. Eksperimenti postojećih metoda

Za eksperimente postojećih metoda preuzete su implementacije iz službenih GitHub repozitorija autora. Metode su pokretane prema dostupnim uputama, pri čemu su uglavnom korišteni zadani parametri ili postavke preporučene u radu [23]. Cilj eksperimenata bio je procijeniti ponašanje metoda u standardiziranim uvjetima i osigurati usporedivost rezultata. Kako bi se dobio potpuniji uvid u ponašanje klasifikatora, izvorni kod svake metode nadopunjen je izračunom mjere F1 i matrice zabune. Sve metode evaluirane su na prethodno opisanom skupu za testiranje.

Metoda Universal Fake Detect dodatno je naučena na LDM skupu kako bi se procijenila njezina sposobnost generalizacije pri učenju na isključivo difuzijskim generiranim slikama.

Važno je napomenuti da usporedbe s radom [23] nisu izravne jer su autori koristili skupove isključivo generiranih i isključivo stvarnih slika. Stoga, kada se u ovom radu uspoređuje točnost na nekom generativnom modelu s njihovom, zapravo se uspoređuje s točnošću postignutom na skupu isključivo generiranih slika tog modela, bez uključivanja stvarnih slika kao što je slučaj u ovom radu.

5.1. Rezultati

5.1.1. CNNDetection

Metoda CNNDetection testirana je s dvije vrijednosti vjerojatnosti primjene augmentacija zamućenja i JPEG kompresije, kao u originalnom radu: 0.5 i 0.1. Rezultati su prikazani u tablicama 5.1. i 5.2. Na GAN-generiranim slikama rezultati odgovaraju onima iz originalnog rada, pri čemu metoda postiže bolje rezultate s vjerojatnošću augmenta-

cije 0.1. Na slikama difuzijskih, transformerskih i modela nepoznate arhitekture obje varijante generaliziraju podjednako loše, s točnošću bliskom slučajnom klasifikatoru.

Tablica 5.1. Rezultati CNNDetection metode s vjerojatnošću augmentacije 0.5

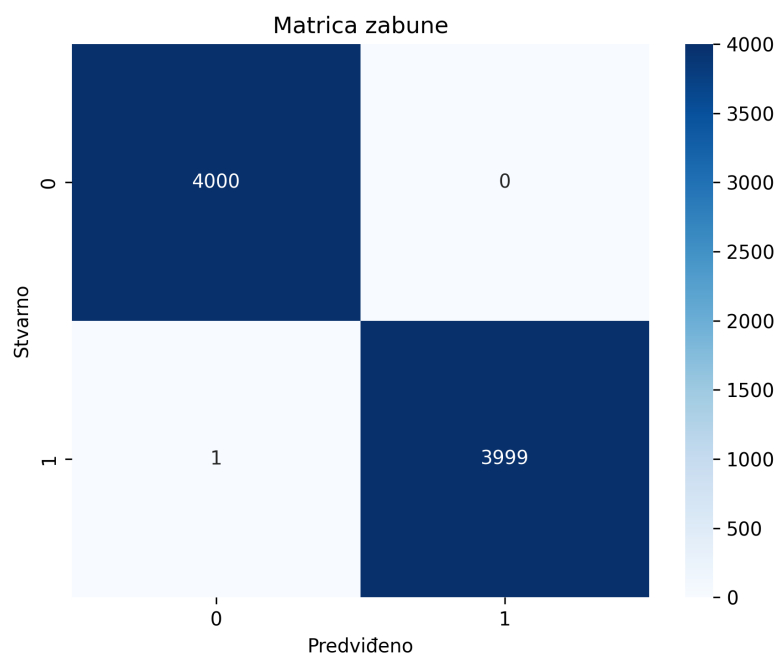
Vrsta	Model	Točnost (%)	F1
GAN	ProGAN	100	1
	BigGAN	58.98	0.3154
	CycleGAN	80.77	0.7659
	GauGAN	79.28	0.7406
	StarGAN	80.94	0.7665
	StyleGAN	73.41	0.638
	StyleGAN2	68.43	0.5388
Difuzijski	SD-2.1	50.08	0.0091
	SD-3	50.70	0.0332
	SD-3.5	50.54	0.0271
	SDXL	50.36	0.0201
Transformer	DALL-E3	49.87	0.0004
Nepoznato	Midjourney	49.96	0.0042

Tablica 5.2. Rezultati CNNDetection metode s vjerojatnošću augmentacije 0.1

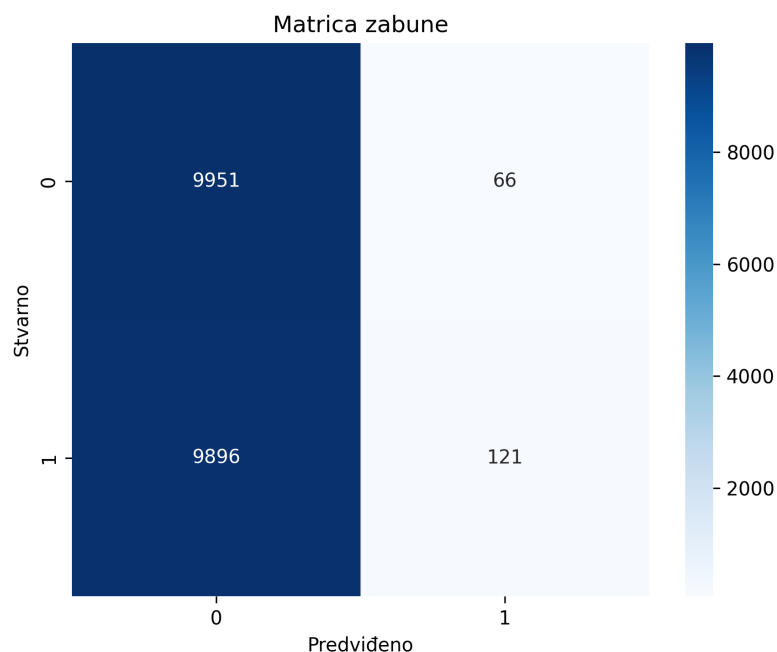
Vrsta	Model	Točnost (%)	F1
GAN	ProGAN	100	1
	BigGAN	70.18	0.611
	CycleGAN	85.24	0.8424
	GauGAN	78.93	0.7547
	StarGAN	91.70	0.9125
	StyleGAN	87.09	0.8519
	StyleGAN2	84.39	0.8151
Difuzijski	SD-2.1	50.27	0.0237
	SD-3	51.11	0.0557
	SD-3.5	50.66	0.039
	SDXL	50.64	0.0379
Transformer	DALL-E3	49.72	0.0018
Nepoznato	Midjourney	49.91	0.0095

Iz rezultata i matrice zabune na slici 5.1. vidljivo je da metoda savršeno detektira ProGAN slike, što je očekivano budući da je upravo na tom modelu učena. Na ostalim GAN modelima točnost varira, pri čemu BigGAN postiže najslabije rezultate zbog strukturalne razlike u odnosu na ProGAN. Na slikama difuzijskih, transformerskih i modela nepoznate arhitekture točnost je bliska 50%, no mjera F1 bliska nuli pokazuje da metoda

gotovo sve takve slike pogrešno klasificira kao stvarne, što je vidljivo u matrici zabune na slici 5.2. Smanjenje vjerojatnosti augmentacije s 0.5 na 0.1 značajno poboljšava rezultate na GAN modelima, no na ostalim arhitekturama ne donosi poboljšanje, što potvrđuje da je problem generalizacije strukturalne prirode, a ne posljedica odabira hiperparametara.



Slika 5.1. Matrica zabune CNNDetection metode s vjerojatnošću augmentacije 0.1 za ProGAN: gotovo savršena klasifikacija



Slika 5.2. Matrica zabune CNNDetection s vjerojatnošću augmentacije 0.1 metode za Stable Diffusion 2.1: gotovo sve slike klasificirane su kao stvarne (oznaka 0)

Autori rada [23] izvještavaju o nešto boljim rezultatima metode CNNDetection na difuzijskim modelima, no u tom su radu korištene ranije verzije difuzijskih modela. Razlika u rezultatima sugerira da novije verzije generiraju realističnije slike koje je teže detektirati, što je u skladu s općim trendom napretka generativnih modela.

5.1.2. PatchCraft

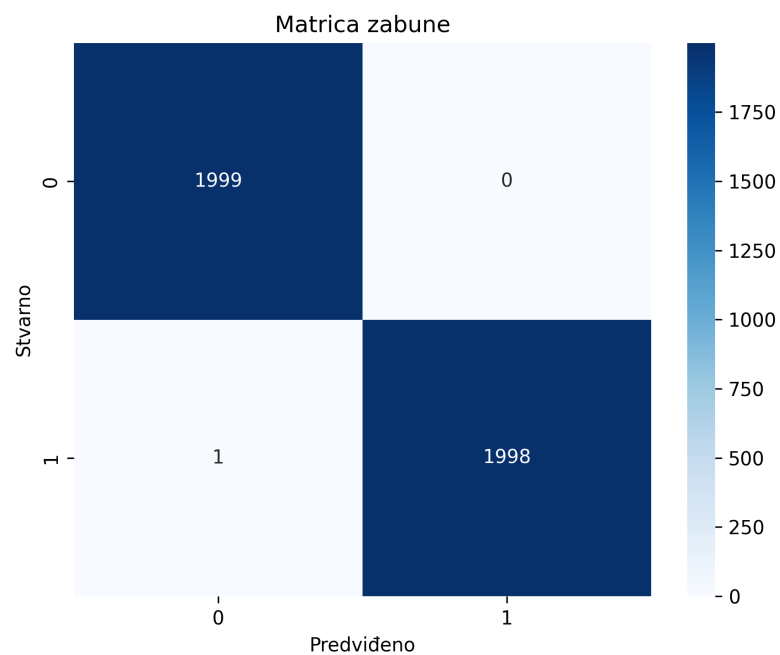
Rezultati PatchCraft metode prikazani su u tablici 5.3. Metoda pokazuje znatno bolju sposobnost generalizacije od metode CNNDetection. Na GAN-ovima su rezultati približno jednaki onima iz originalnog rada, no na difuzijskim, transformerskim i nepoznatim arhitekturama su lošiji. Takvi rezultati vjerojatno su posljedica korištenja novijih verzija tih modela koje je teže detektirati.

Tablica 5.3. Rezultati PatchCraft metode

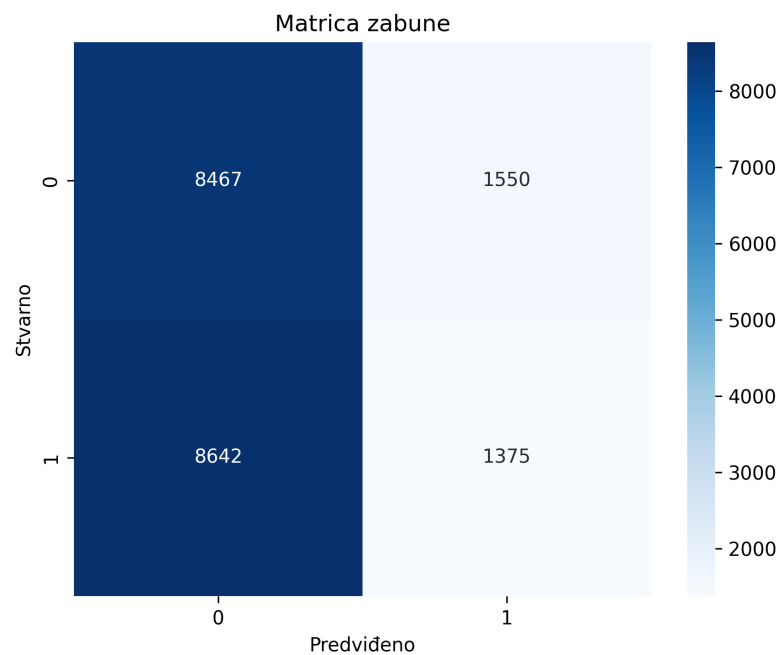
Vrsta	Model	Točnost (%)	F1
GAN	ProGAN	100	1
	BigGAN	95.6	0.9568
	CycleGAN	69.34	0.6002
	GauGAN	71.88	0.6739
	StarGAN	99.97	0.9997
	StyleGAN	92.8	0.9303
	StyleGAN2	89.49	0.8884
Difuzijski	SD-2.1	59.47	0.4589
	SD-3	49.13	0.2125
	SD-3.5	57.07	0.4093
	SDXL	65.96	0.5823
Transformer	DALL-E3	52.03	0.2125
Nepoznato	Midjourney	68.42	0.6242

Na većini GAN modela metoda postiže visoku točnost, uz iznimku CycleGAN i GauGAN modela koji su slabije detektirani. Razlog tomu je vjerojatno strukturalna razlika njihovih artefakata. Na difuzijskim modelima rezultati su promjenjivi. SDXL je detektiran s točnošću od 65.96%, dok SD-3 postiže svega 49.13%, što ukazuje na to da različite verzije iste arhitekture mogu proizvoditi artefakte različite detekcijske prepoznatljivosti. Metoda postiže iznenađujuće dobre rezultate na Midjourney slikama unatoč nepoznatoj arhitekturi tog modela, što sugerira da teksturni artefakti koje PatchCraft detektira nisu strogo specifični za jednu familiju generativnih modela.

Slika 5.3. prikazuje matricu zabune za StarGAN kao primjer savršene klasifikacije, a slika 5.4. prikazuje matricu zabune za StableDiffusion 3, koji je najlošije detektiran ovom metodom.



Slika 5.3. Matrica zabune za metodu PatchCraft za StarGAN



Slika 5.4. Matrica zabune za metodu PatchCraft za StableDiffusion 3

5.1.3. Universal Fake Detect

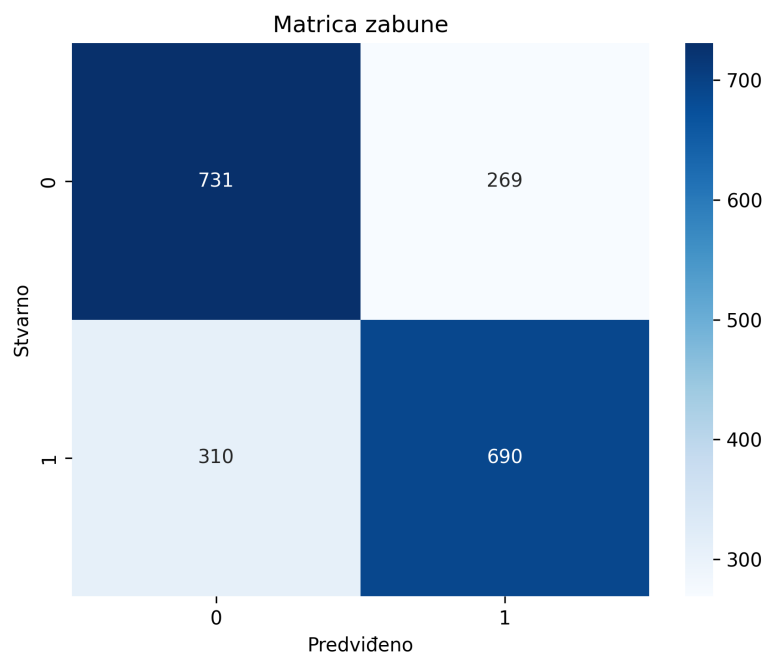
Rezultati Universal Fake Detect metode prikazani su u tablici 5.4. Na GAN-ovima su rezultati približno jednaki onima iz originalnog rada. Na difuzijskim i transformerskim modelima rezultati su nešto slabiji od onih iz rada [23], što je vjerojatno posljedica korištenja novijih verzija modela koje je teže detektirati, kao što je pokazano i kod prethodnih metoda.

Tablica 5.4. Rezultati Universal Fake Detect metode

Vrsta	Model	Točnost (%)	F1
GAN	ProGAN	99.92	1.0
	BigGAN	96.25	0.96
	CycleGAN	98.42	0.98
	GauGAN	99.6	1.0
	StarGAN	96.4	0.96
	StyleGAN	90.33	0.9
	StyleGAN2	92.92	0.93
Difuzijski	SD-2.1	75.1	0.75
	SD-3	71.0	0.7
	SD-3.5	73.5	0.73
	SDXL	70.0	0.69
Transformer	DALL-E3	49.95	0.67
Nepoznato	Midjourney	54.3	0.64

Universal Fake Detect postiže izvrsne rezultate na GAN modelima, s točnošću iznad 90% za sve testirane modele, što predstavlja poboljšanje u odnosu na PatchCraft na CycleGAN i GauGAN modelima. Na difuzijskim modelima metoda generalizira znatno bolje od CNNDetection i PatchCrafta, postižući točnost između 70% i 75%, što potvrđuje da CLIP reprezentacije hvataju općenitije značajke neovisne o specifičnoj arhitekturi generativnog modela. Iznimka je DALL-E 3, gdje točnost od 49.95% uz mjeru F1 0.67 ukazuje na neuravnoteženu klasifikaciju. Metoda detektira generirane slike, no uz visok udio lažnih pozitiva. Slični rezultati vidljivi su i za Midjourney. Na slici 5.5. prikazana je matrica zabune za Stable Diffusion 3, što omogućuje usporedbu s rezultatima metode PatchCraft na istom modelu.

Tablica 5.5. prikazuje rezultate Universal Fake Detect metode učene na skupu sa slikama generiranim isključivo LDM-om u pet epoha. Metoda ostaje uspješna na GAN-ovima unatoč tome što nije učena na modelima iz te familije, što potvrđuje dobru spo-



Slika 5.5. Matrica zabune za Universal Fake Detect za Stable Diffusion 3

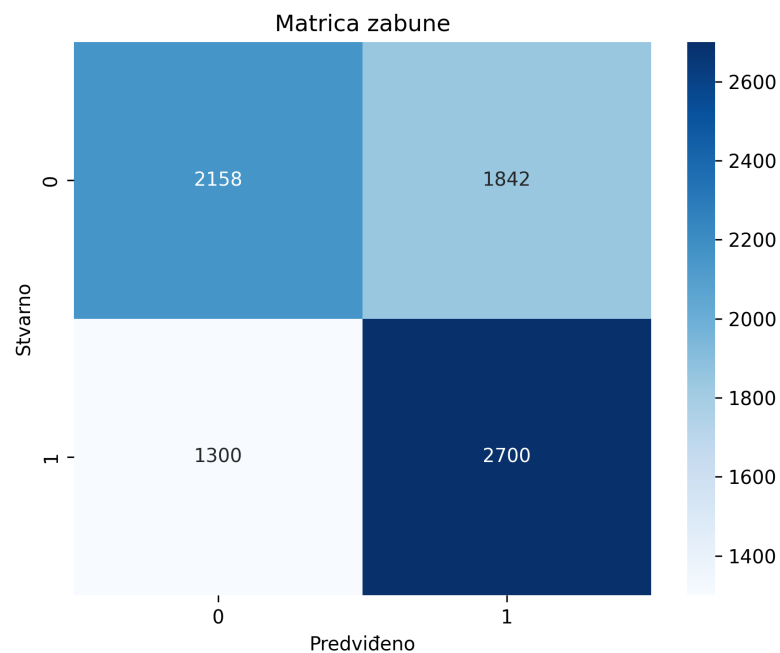
sobnost generalizacije. Iznimka su StyleGAN i StyleGAN2, koji su detektirani s točnošću manjom od 90%, za razliku od varijante učenje na ProGAN-u gdje su ti modeli detektirani s višom točnošću. To sugerira da ProGAN i StyleGAN dijele određene zajedničke artefakte koje metoda učenja na ProGAN-u lakše prepoznaje. Slika 5.6. prikazuje matricu zabune za StyleGAN2, koji je najlošije detektiran ovom metodom.

Na difuzijskim modelima rezultati su nešto slabiji nego u varijanti učenju na ProGAN-u, što je očekivano s obzirom na to da su difuzijske slike korištene u učenju generirane isključivo LDM-om, dok testni skup uključuje novije i raznovrsnije varijante. Međutim, na transformerskoj arhitekturi dolazi do značajnog poboljšanja, kao i na Midjourney slikama. Točnost na DALL-E 3 raste s 49.95% na 83.55%, a na Midjourney slikama s 54.3% na 71.35%. Matrica zabune za DALL-E 3 prikazana je na slici 5.7.

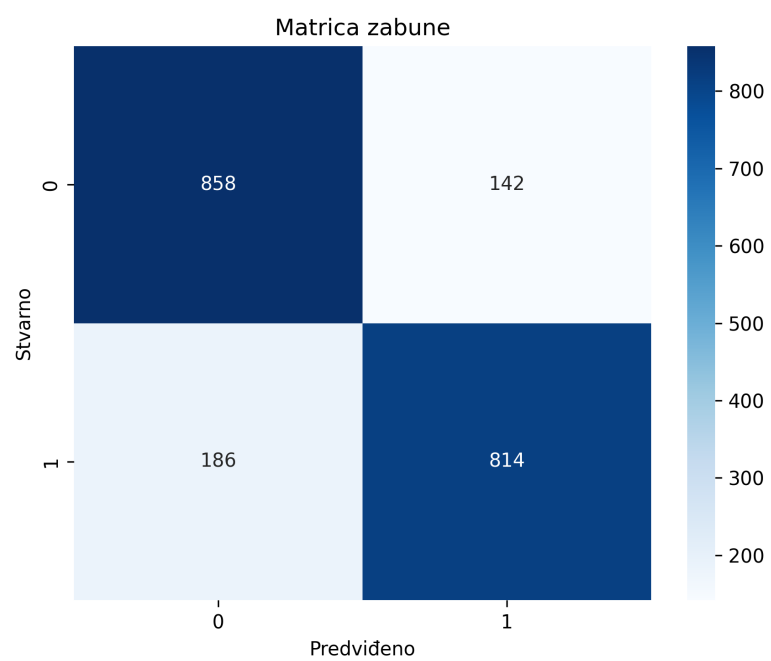
Iz navedenog proizlazi da metoda bolje generalizira na različite generativne modele kada je učenje na LDM slikama nego na ProGAN slikama. To sugerira da difuzijski modeli generiraju raznovrsnije i kompleksnije artefakte od GAN-ova, zbog čega model učen na takvim slikama nauči općenitije značajke generiranih slika neovisne o specifičnom generativnom modelu.

Tablica 5.5. Rezultati Universal Fake Detect metode ućene na LDM skupu

Vrsta	Model	Toćnost (%)	F1
GAN	ProGAN	92.75	0.93
	BigGAN	92.2	0.92
	CycleGAN	94.67	0.95
	GauGAN	98.6	0.99
	StarGAN	92.1	0.92
	StyleGAN	75.68	0.74
	StyleGAN2	60.71	0.63
Difuzijski	SD-2.1	72.55	0.72
	SD-3	71.05	0.7
	SD-3.5	66.6	0.65
	SDXL	67.45	0.66
Transformer	DALL-E3	83.55	0.83
Nepoznato	Midjourney	71.35	0.71



Slika 5.6. Matrica zabune za ćuniversal Fake Detect ućen na LDM skupu za StyleGAN2



Slika 5.7. Matrica zabune za Universal Fake Detect učen na LDM skupu za DALL-E 3

6. Adaptivna fuzija CNNDetection i Universal Fake Detect metode

Nakon analize rezultata postojećih metoda, ovaj rad kao novu metodu predlaže model dvograne arhitekture koja kombinira značajke CNNDetection i Universal Fake Detect metode.

Motivacija za takvu kombinaciju proizlazi iz komplementarnih svojstava dviju metoda. CNNDetection postiže visoku točnost na GAN-generiranim slikama, no gotovo potpuno zakazuje na difuzijskim i transformerskim modelima. S druge strane, Universal Fake Detect učen na LDM slikama postiže dobru generalizaciju na difuzijske i transformerske modele, no nešto slabije rezultate na određenim GAN modelima poput StyleGAN i StyleGAN2. Kombinacijom značajki obiju metoda cilj je izgraditi detektor koji pokriva širi spektar generativnih modela i postiže bolju ukupnu generalizaciju od svake metode zasebno.

6.1. Arhitektura

Arhitektura predložene metode temelji se na spajanju CNNDetection i Universal Fake Detect metode u jedinstveni model dvograne arhitekture, gdje svaka grana odgovara jednoj od tih metoda. Svaka grana iz ulazne slike izvlači značajke karakteristične za tu metodu, koje se zatim kombiniraju dodatnom mrežom za konačnu klasifikaciju.

Budući da svaka grana očekuje ulaznu sliku predobrađenu na specifičan način, ista se slika zasebno predobrađuje za svaku granu prema parametrima iz originalnih radova. CNNDetection grana očekuje ulaz normaliziran ImageNetovom normalizacijom, dok Universal Fake Detect grana očekuje CLIP-ovu normalizaciju.

Modeli okosnica obiju grana zamrznuti su tijekom učenja, pri čemu se učitavaju pret-

hodno naučene težine iz originalnih implementacija ili naučene na dodatnom skupu. Budući da cilj nije reproducirati originalnu klasifikaciju pojedinih metoda već iskoristiti njihove naučene reprezentacije, zadnji klasifikacijski sloj uklanja se iz obje okosnice i zamjenjuje slojem identiteta. Time se kao izlaz svake grane dobiva vektor značajki: vektor dimenzije 2048 iz ResNet50 okosnice CnNDetection grane i vektor dimenzije 768 iz ViT-L/14 okosnice Universal Fake Detect grane.

Budući da su izlazni vektori različitih dimenzija, ne mogu se izravno spajati. Stoga se oba vektora linearnom projekcijom preslikavaju u zajednički prostor dimenzije 512, nakon čega se primjenjuje GELU aktivacijska funkcija. Fuzija značajki odnosi se na kombiniranje reprezentacija iz više izvora u jedinstvenu reprezentaciju pogodnu za klasifikaciju. GELU aktivacija preferira se nad ReLU-om u takvim problemima jer omogućuje mekši prijelaz vrijednosti, što pogoduje učenju u situacijama kada značajke dolaze iz različitih izvora.

Ključni dio arhitekture je mehanizam adaptivnog određivanja težina koji dinamički određuje koliko se treba osloniti na značajke svake grane za pojedinu ulaznu sliku. Konkatencija projiciranih vektora dovodi se na linearni sloj čiji izlaz prolazi kroz softmax funkciju, čime se dobivaju dvije težine α_1 i α_2 koje zadovoljavaju uvjet $\alpha_1 + \alpha_2 = 1$. Kombinirani vektor značajki tada se računa kao

$$f = \alpha_1 p_1 + \alpha_2 p_2 \quad (6.1)$$

gdje su p_1 i p_2 projicirani vektori CNN i UFD grane. Za razliku od jednostavnog prosjeka, ovaj mehanizam omogućuje modelu da za svaku sliku zasebno procijeni pouzdanost svake grane, što je posebno korisno jer se očekuje da će CNN grana biti pouzdanija za GAN-generirane slike, a UFD grana za difuzijske i transformerske modele.

Kombinirani vektor značajki dovodi se na završni linearni klasifikacijski sloj čiji izlaz prolazi kroz sigmoidnu aktivacijsku funkciju, čime se dobiva vjerojatnost da je slika detektirana u intervalu $[0, 1]$. Pri učenju se optimira gubitak binarne unakrsne entropije, a prag klasifikacije optimira se na skupu za validaciju maksimizacijom mjere F1.

6.2. Implementacija

Model je implementiran u programskom jeziku Python korištenjem biblioteke PyTorch. Za učitavanje i predobradu slika koristi se biblioteka torchvision, a za evaluaciju metrika biblioteka scikit-learn.

Ključni dio implementacije je klasa `FusionHead` koja implementira mehanizam fuzije značajki i adaptivnog određivanja težina. Prikazana je u isječku koda 6.1

Kod 6.1: Klasa `FusionHead`

```
class FusionHead(nn.Module):  
    def __init__(self):  
        super().__init__()  
        self.proj_cnn = nn.Linear(2048, 512)  
        self.proj_clip = nn.Linear(768, 512)  
        self.gate = nn.Linear(1024, 2)  
        self.classifier = nn.Linear(512, 1)  
  
    def forward(self, f_cnn, f_clip):  
        p1 = F.gelu(self.proj_cnn(f_cnn))  
        p2 = F.gelu(self.proj_clip(f_clip))  
        alpha = F.softmax(  
            self.gate(torch.cat([p1, p2], dim=-1)),  
            dim=-1  
        )  
        fused = alpha[:, 0:1] * p1 + alpha[:, 1:2] * p2  
        logit = self.classifier(fused).squeeze(1)  
        return logit, alpha
```

Modeli okosnica učitavaju se s prethodno naučenim težinama i zamrzavaju postavljanjem `requires_grad = False` na sve parametre. Zadnji klasifikacijski sloj svake okosnice zamjenjuje se slojem identiteta kako bi se kao izlaz dobio vektor značajki umjesto klasifikacijske odluke.

Model se uči kroz 5 epoha s veličinom batcha 64 korištenjem AdamW optimizatora sa

stopom učenja 10^{-4} . Kao funkcija gubitka koristi se gubitak binarne unakrsne entropije s ugrađenom sigmoidnom aktivacijom (`BCEWithLogitsLoss`). Jedini parametri koji se optimiraju su parametri klase `FusionHead`: projekcijski slojevi, mehanizam vrata (engl. *gate*) i klasifikator. Okosnice ostaju zamrznute.

Nakon svake epohe na skupu za validaciju optimira se prag klasifikacije maksimizacijom mjere F1 pretraživanjem 200 ravnomjerno raspoređenih vrijednosti u rasponu $[0, 1]$, što je prikazano u isječku koda 6.2. Model s najboljom mjerom F1 na validacijskom skupu sprema se zajedno s odgovarajućim pragom.

Kod 6.2: Funkcija za određivanje praga klasifikacije

```
def find_best_threshold(y_true, y_prob):
    thresholds = np.linspace(0, 1, 200)
    best_t, best_f1 = 0.5, 0
    for t in thresholds:
        pred = (y_prob >= t).astype(int)
        f1 = f1_score(y_true, pred)
        if f1 > best_f1:
            best_f1 = f1
            best_t = t
    return best_t
```

6.3. Rezultati

Tablica 6.1. prikazuje rezultate Adaptivne fuzije `CNNDetection` i `Universal Fake Detect` metode, pri čemu je `Universal Fake Detect` prethodno naučen na `ProGAN` skupu, kao u originalnom radu te metode.

Na GAN slikama metoda postiže nešto slabije rezultate od najboljih analiziranih postojećih metoda, no razlike su relativno male, posebno na `BigGAN` i `GauGAN` skupovima. Najveći pad vidljiv je na `StyleGAN2`, što upućuje na poteškoće generalizacije na novije GAN arhitekture.

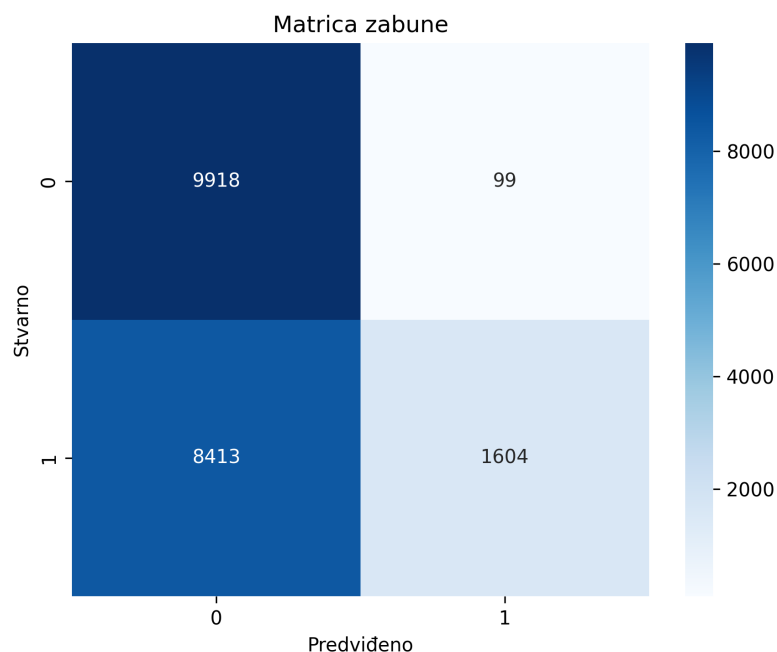
Na difuzijskim modelima metoda nadmašuje `CNNDetection` i `PatchCraft`, no nešto zaostaje za originalnom `Universal Fake Detect` metodom. Pritom se uočava trend pada

točnosti s novijim verzijama Stable Diffusion modela, što sugerira da noviji difuzijski modeli predstavljaju veći izazov za detekciju.

Na transformerskoj arhitekturi metoda postiže bolje rezultate od svih originalnih analiziranih metoda, dok je na Midjourneyju neznatno slabija od PatchCraft metode. Matrica zabune za DALL-E 3 prikazana je na slici 6.1.

Tablica 6.1. Rezultati Adaptivne fuzije CNNDetection i Universal Fake Detect metode s Universal Fake Detectom prethodno naučenim na ProGAN skupu

Vrsta	Model	Točnost (%)	F1
GAN	ProGAN	87.35	0.8874
	BigGAN	92.73	0.9246
	CycleGAN	91.71	0.9231
	GauGAN	94.27	0.9457
	StarGAN	81.49	0.8236
	StyleGAN	80.68	0.81
	StyleGAN2	74.1	0.716
Difuzijski	SD-2.1	70.05	0.5783
	SD-3	62.21	0.4021
	SD-3.5	58.07	0.2899
	SDXL	63.83	0.4419
Transformer	DALL-E3	57.51	0.2737
Nepoznato	Midjourney	65.94	0.4911



Slika 6.1. Matrica zabune za Adaptivnu fuziju s Universal Fake Detectom prethodno naučenim na ProGAN skupu

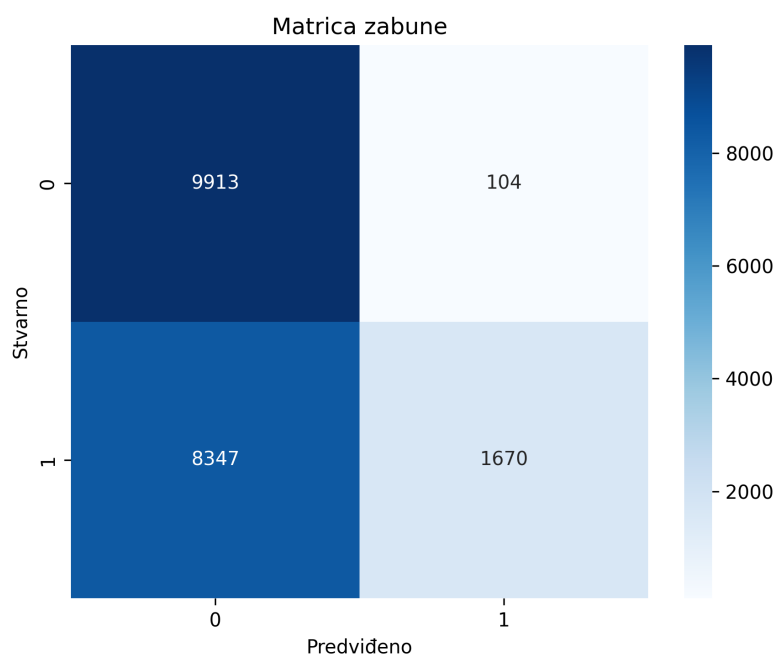
Tablica 6.2. prikazuje rezultate iste metode s Universal Fake Detectom naučenim na LDM skupu. Rezultati su konzistentno nešto bolji u svim kategorijama u usporedbi s ProGAN varijantom, što sugerira da LDM skup pruža korisnije reprezentacije za generalizaciju. Poboljšanja su posebno vidljiva na StarGAN i Midjourney skupovima, dok na ostalim skupovima razlike ostaju minimalne. Na slici 6.2. prikazana je matrica zabune za DALL-E 3, za usporedbu s istom metodom s Universal Fake Detectom naučenom na ProGAN skupu.

Tablica 6.3. prikazuje točnosti svih analiziranih metoda. CNNDet označava metodu CNNDetection zajedno s pripadajućim vjerojatnostima augmentacije, Patch metodu PatchCraft, UFD metodu Universal Fake Detect prethodno naučenu na ProGAN-u, odnosno LDM-u, a Fuzija metodu Adaptivne fuzije, gdje ProGAN, odnosno LDM, označavaju generativni model na kojem je naučen Universal Fake Detect.

PatchCraft postiže bolje rezultate na difuzijskim i transformerskim arhitekturama od metode CNNDetection, zbog čega bi u budućem radu bilo zanimljivo isprobati fuziju PatchCraft i Universal Fake Detect metode. Takva kombinacija potencijalno bi mogla postići bolju generalizaciju na novije generativne arhitekture, no zbog računalnih zahtjeva obiju metoda ta kombinacija nije razmatrana u ovom radu.

Tablica 6.2. Rezultati Adaptivne fuzije CNNDetection i Universal Fake Detect metode s Universal Fake Detectom prethodno naučenim na LDM skupu

Vrsta	Model	Točnost (%)	F1
GAN	ProGAN	88.03	0.8927
	BigGAN	92.98	0.9269
	CycleGAN	92.39	0.9292
	GauGAN	94.36	0.9465
	StarGAN	84.62	0.8557
	StyleGAN	81.46	0.8142
	StyleGAN2	74.49	0.7229
Difuzijski	SD-2.1	70.86	0.5948
	SD-3	62.73	0.4156
	SD-3.5	58.65	0.3071
	SDXL	64.17	0.4504
Transformer	DALL-E3	57.82	0.2833
Nepoznato	Midjourney	66.98	0.5145



Slika 6.2. Matrica zabune za adaptivnu fuziju s Universal Fake Detectom prethodno naučenim na LDM skupu

Tablica 6.3. Usporedba točnosti svih analiziranih metoda

Vrsta	Model	CNNDet		Patch	UFD		Fuzija	
		0.1	0.5		ProGAN	LDM	ProGAN	LDM
GAN	ProGAN	100	100	100	99.92	92.75	87.35	88.03
	BigGAN	70.18	58.98	95.6	96.25	92.2	92.73	92.98
	CycleGAN	85.24	80.77	69.34	98.42	94.67	91.71	92.39
	GauGAN	78.93	79.28	71.88	99.6	98.6	94.24	94.36
	StarGAN	91.70	80.94	99.97	96.4	92.1	81.49	84.62
	StyleGAN	87.09	73.41	92.8	90.33	75.68	80.68	81.46
	StyleGAN2	84.39	68.43	89.49	92.92	60.71	74.1	74.49
Difuzijski	SD-2.1	50.27	50.08	59.47	75.1	72.55	70.05	70.86
	SD-3	51.11	50.70	49.13	71.0	71.05	62.21	62.73
	SD-3.5	50.66	50.54	57.07	73.5	66.6	58.07	58.65
	SDXL	50.64	50.36	65.96	70.0	67.45	63.83	64.17
Transformer	DALL-E3	49.72	49.87	52.03	49.95	83.55	57.51	57.82
Nepoznato	Midjourney	49.91	49.96	68.42	54.3	71.35	65.94	66.98

7. Zaključak

U ovom radu analizirane su i uspoređene tri postojeće metode detekcije umjetno generiranih slika, CNNDetection, PatchCraft i Universal Fake Detect, te je predložena nova metoda Adaptivne fuzije koja kombinira CNNDetection i Universal Fake Detect.

Rezultati pokazuju da nijedna od metoda ne postiže jednaku uspješnost na svim vrstama generativnih modela. Universal Fake Detect postiže najbolje rezultate na GAN slikama, posebno kada je naučen na ProGAN skupu, dok CNNDetection s pragom 0.1 ostvaruje vrlo dobre rezultate na GAN-ovima, ali slabije generalizira na novije arhitekture. PatchCraft pokazuje najstabilnije ponašanje kroz različite kategorije podataka te zadržava razumnu točnost i na difuzijskim modelima, Midjourneyju i transformerskoj arhitekturi.

Predložena metoda Adaptivne fuzije postiže konkurentne rezultate i potvrđuje da kombiniranje dviju metoda može biti korisno za detekciju na novijim generativnim arhitekturama. Na GAN slikama neznatno zaostaje za Universal Fake Detectom, dok na difuzijskim i transformerskim modelima nadmašuje CNNDetection te postiže najbolje rezultate među analiziranim metodama na DALL-E 3. Verzija s Universal Fake Detectom naučenim na LDM skupu pokazuje nešto konzistentnije rezultate od varijante naučene na ProGAN skupu.

Glavni zaključak rada jest da generalizacija na neviđene generativne modele ostaje otvoren problem. Metode koje postižu visoku točnost na poznatim arhitekturama često slabo prenose znanje na novije generatore, što potvrđuje potrebu za razvojem robusnijih pristupa detekciji. U budućem radu bilo bi korisno istražiti naprednije strategije fuzije, uključiti dodatne metode u kombinaciju i detaljnije analizirati značajke koje doprinose boljoj generalizaciji među različitim generativnim arhitekturama.

Literatura

- [1] K. Vass, “Harold cohen aaron - computer art”, Dec 2022. [Mrežno]. Adresa: <https://www.katevassgalerie.com/blog/harold-cohen-aaron-computer-art>
- [2] Midjourney, Inc., “Midjourney”, Artificial Intelligence System, 2022. [Mrežno]. Adresa: <https://midjourney.com>
- [3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, i Y. Bengio, “Generative adversarial nets”, *Advances in neural information processing systems*, sv. 27, 2014.
- [4] M. Mirza i S. Osindero, “Conditional generative adversarial nets”, *arXiv preprint arXiv:1411.1784*, 2014.
- [5] T. Karras, T. Aila, S. Laine, i J. Lehtinen, “Progressive growing of gans for improved quality, stability, and variation”, *arXiv preprint arXiv:1710.10196*, 2017.
- [6] A. Brock, J. Donahue, i K. Simonyan, “Large scale gan training for high fidelity natural image synthesis”, *arXiv preprint arXiv:1809.11096*, 2018.
- [7] T. Karras, S. Laine, i T. Aila, “A style-based generator architecture for generative adversarial networks”, u *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019., str. 4401–4410.
- [8] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, i T. Aila, “Analyzing and improving the image quality of stylegan”, u *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020., str. 8110–8119.

- [9] J.-Y. Zhu, T. Park, P. Isola, i A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks”, u *Proceedings of the IEEE international conference on computer vision*, 2017., str. 2223–2232.
- [10] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, i J. Choo, “Stargan: Unified generative adversarial networks for multi-domain image-to-image translation”, u *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018., str. 8789–8797.
- [11] T. Park, M.-Y. Liu, T.-C. Wang, i J.-Y. Zhu, “Semantic image synthesis with spatially-adaptive normalization”, u *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019., str. 2337–2346.
- [12] J. Ho, A. Jain, i P. Abbeel, “Denoising diffusion probabilistic models”, *Advances in neural information processing systems*, sv. 33, str. 6840–6851, 2020.
- [13] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, i B. Ommer, “High-resolution image synthesis with latent diffusion models”, u *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022., str. 10 684–10 695.
- [14] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, i R. Rombach, “Sdxl: Improving latent diffusion models for high-resolution image synthesis”, u *International Conference on Learning Representations*, sv. 2024, 2024., str. 1862–1874.
- [15] I. Admin, “Introducing stable diffusion 3.5”, Nov 2024. [Mrežno]. Adresa: <https://stability.ai/news-updates/introducing-stable-diffusion-3-5>
- [16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, i I. Polosukhin, “Attention is all you need”, *Advances in neural information processing systems*, sv. 30, 2017.
- [17] OpenAI, “Improving image generation with better captions”, OpenAI, teh. izv., 2023. [Mrežno]. Adresa: <https://cdn.openai.com/papers/dall-e-3.pdf>
- [18] R. Yamashita, M. Nishio, R. K. G. Do, i K. Togashi, “Convolutional neural networks: an overview and application in radiology”, *Insights into imaging*, sv. 9, br. 4, str. 611–

629, 2018.

- [19] K. He, X. Zhang, S. Ren, i J. Sun, “Deep residual learning for image recognition”, u *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016., str. 770–778.
- [20] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision”, u *International conference on machine learning*. PmLR, 2021., str. 8748–8763.
- [21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale”, *arXiv preprint arXiv:2010.11929*, 2020.
- [22] U. Ojha, Y. Li, i Y. J. Lee, “Towards universal fake image detectors that generalize across generative models”, u *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023., str. 24 480–24 489.
- [23] D. Park, H. Na, i D. Choi, “Performance comparison and visualization of ai-generated-image detection methods”, *IEEE Access*, sv. 12, str. 62 609–62 627, 2024.
- [24] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, i A. A. Efros, “Cnn-generated images are surprisingly easy to spot... for now”, u *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020., str. 8695–8704.
- [25] N. Zhong, Y. Xu, S. Li, Z. Qian, i X. Zhang, “Patchcraft: Exploring texture patch for efficient ai-generated image detection”, *arXiv preprint arXiv:2311.12397*, 2023.
- [26] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, i J. Xiao, “Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop”, *arXiv preprint arXiv:1506.03365*, 2015.
- [27] 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 (CVPR)*. IEEE, 2009., str. 248–255.

- [28] R. Corvi, D. Cozzolino, G. Zingarini, G. Poggi, K. Nagano, i L. Verdoliva, “On the detection of synthetic images generated by diffusion models”, u *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023., str. 1–5. <https://doi.org/10.1109/ICASSP49357.2023.10095167>
- [29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, i C. L. Zitnick, “Microsoft coco: Common objects in context”, u *European conference on computer vision*. Springer, 2014., str. 740–755.
- [30] Z. Liu, P. Luo, X. Wang, i X. Tang, “Deep learning face attributes in the wild”, u *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.
- [31] N. Imanpour, S. Bajpai, S. Ghosh, S. R. Sankepally, A. Borah, H. M. Abdullah, N. Kosaraju, S. Dixit, A. Aziz, S. Biswas, V. Jain, A. Chadha, A. Sheth, i A. Das, “Visual counter turing test (VCT²): Discovering the challenges for ai-generated image detection and introducing visual ai index (v_ai)”, 2024. [Mrežno]. Adresa: <https://arxiv.org/abs/2411.16754>

Sažetak

Detekcija umjetno generiranih slika primjenom metoda strojnog i dubokog učenja

Josipa Sever

Ovaj rad istražuje problem detekcije umjetno generiranih slika, s posebnim naglaskom na generalizaciju na neviđene generativne modele. Analizirane su tri postojeće metode, CNNDetection, PatchCraft i Universal Fake Detect, koje koriste tehnike dubokog i strojnog učenja. Metode su evaluirane na skupu slika generiranih različitim arhitekturama, uključujući GAN modele, difuzijske modele i transformere. Rezultati pokazuju da nijedna metoda ne generalizira jednako dobro na sve vrste generativnih modela: Universal Fake Detect postiže najbolje rezultate na GAN slikama, dok PatchCraft pokazuje uravnoteženije ponašanje na skupovima temeljenima na difuzijskim i transformerskim modelima. Dodatno, predložena je i implementirana nova metoda Adaptivne fuzije koja kombinira CNNDetection i Universal Fake Detect. Evaluacija pokazuje da predložena metoda nadmašuje sve analizirane metode na transformerskoj arhitekturi, dok na GAN slikama i difuzijskim modelima postiže usporedive rezultate s metodama od kojih je sastavljena.

Ključne riječi: generativni modeli, strojno učenje, duboko učenje, računalni vid, detekcija generiranih slika, generalizacija

Abstract

Detection of AI-generated images using machine learning and deep learning methods

Josipa Sever

This thesis investigates the problem of AI-generated image detection, with a particular focus on generalization to unseen generative models. Three existing methods are analyzed: CNNDetection, PatchCraft, and Universal Fake Detect, all of which rely on deep and machine learning techniques. The methods are evaluated on a dataset of images generated by various architectures, including GAN models, diffusion models, and transformers. The results show that no method generalizes equally well across all types of generative models: Universal Fake Detect achieves the best performance on GAN images, while PatchCraft shows the most balanced performance across diffusion- and transformer-based datasets. Additionally, a new method called Adaptive Fusion is proposed and implemented, combining CNNDetection and Universal Fake Detect. The evaluation shows that the proposed method outperforms all analyzed methods on transformer-based architecture, while achieving comparable results to its constituent methods on GAN images and diffusion models.

Keywords: generative models, machine learning, deep learning, computer vision, AI-generated image detection, generalization