

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

Kružni generativni suparnički modeli

Ivan Šego

Voditelj: *prof. dr. sc. Siniša Šegvić*

Zagreb, lipanj 2019.

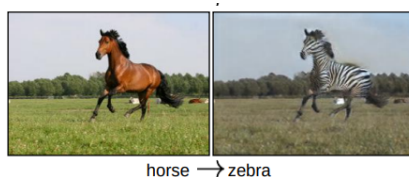
SADRŽAJ

1. Uvod	1
2. Generativni suparnički modeli	2
3. Kružni generativni suparnički modeli	4
3.1. Matematički model	4
3.1.1. Osnovne funkcije gubitka	5
3.1.2. Kružno dosljedna funkcija gubitka	5
4. Arhitektura	7
4.1. Generator	7
4.2. Diskriminator	8
4.3. Treniranje	9
5. Najvažniji rezultati i primjeri	11
6. Zaključak	13
7. Literatura	14
8. Sažetak	15

1. Uvod

Ovaj seminarski rad razmatra generativne suparničke modele te posebice njenu inačicu koju možemo kružni generativni suparnički model (engl. *Cycle Generative Adversarial Network*, *CycleGAN*). U početnim poglavljima se razmatraju općenito generativni suparnički modeli i najbitnije ideje iza njih. U idućim poglavljima se razmatra ideja kružnih GAN-ova koji se prvi puta pojavljuju u Zhu et al. (2017). U seminarskom radu će se se dati pregled i rezultati navedeni u Zhu et al. (2017) te će se istražiti i još poneki konkretni problem koji je moguće riješiti uporabom CycleGAN-a.

Prevođenje iz slike u sliku je zadatak računalnog vida u kojem je ideja transformirati originalnu sliku iz jedne domene u drugu domenu. Primjerice, htjeli bismo iz slike konja dobiti (skoro) identičnu sliku u kojem je konj zamijenjen sa zebrom. To radimo na način da translatiramo iz domene konja u domenu zebri što je vidljivo na slici 1.1. Do sada, većina modela koja bi bila pogodna za ovakve vrste zadataka je zahtijevala uparene slike, što znači da bi se skup podataka sastojao od parova (x, y) npr. slika konja i odgovarajuće zebre. Takve skupove podataka je teško graditi. Zhu et al. (2017) predlaže model u kojem je dovoljno imati dva odvojena skupa, npr. skup slika konja i skupa slika zebre kako bismo dobili rezultate kao što bi bilo u slučaju uparenih podataka.



Slika 1.1: Primjer translacije konja u zebro dobiveno kružnim generativnim suparničkim modelom, preuzeto iz Zhu et al. (2017)

2. Generativni suparnički modeli

Generativni suparnički modeli su prvi puta opisani u Goodfellow et al. (2014). Zadatak GAN-a je generirati umjetne slike koje će biti što sličnije originalnim slikama iz originalnog skupa slika.

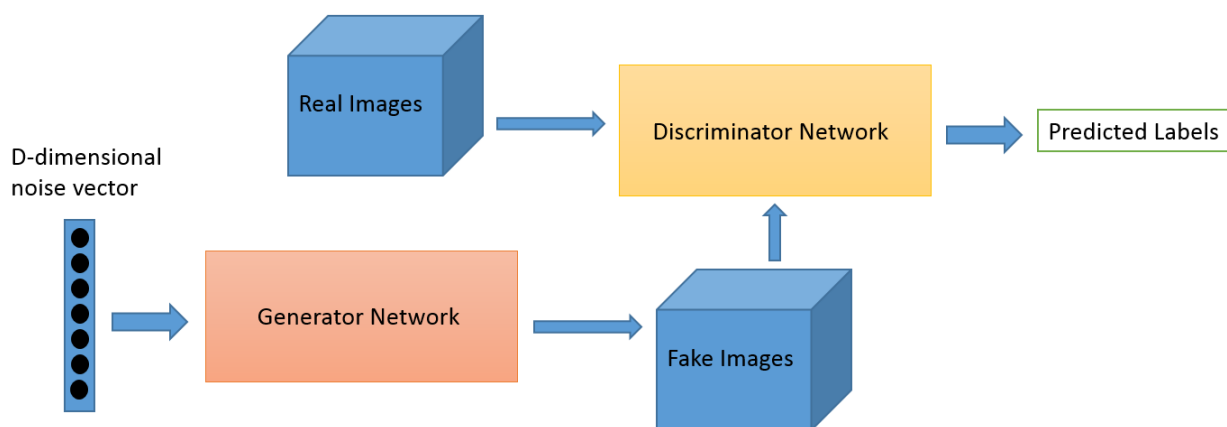
GAN se sastoji od generatora (G) i diskriminatora (D). Prikaz se vidi na slici 2.1

Ideja generatora je da iz nasumično odabranih brojeva generira sliku koja je kompatibilna s distribucijom skupa za učenje. Dakle, ulaz generatora je vektor slučajnih brojeva, a izlaz je slika. Generator možemo matematički zapisati kao $G(z; \theta_g)$ a u praksi ga najčešće predstavljamo konvolucijskim modelima koja modelira pretvorbu varijable (vektora) z u sliku zadanih dimenzija pri čemu su θ_g parametri konvolucijskog modela. Ideja diskriminatora je da razlikuje umjetno generirane slike od onih pravih. Ulaz u diskriminator su slike iz skupa za učenje umjetne slike koje je proizveo generator. Izlaz generatora je vjerojatnost da su slike došle iz originalnog skupa podataka.

Diskriminator možemo matematički zapisati kao $D(x; \theta_d)$, a tipično ga izvodimo konvolucijskim modelom s parametrima θ_d koji na ulazu prima sliku, a na izlazu proizvodi vjerojatnost da je slika stvarna (a ne umjetna). Prema ovakvom opisu problema, cilj diskriminatora je maksimizirati funkciju $D(x)$ te istovremeno minimizirati funkciju $D(G(z))$. Cilj generatora maksimizirati funkciju $D(G(z))$. Ovaj cilj možemo ujediniti u jednu jednadžbu prikazanu u (2.1).

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

U jednadžbi (2.1) prvi pribrojnik predstavlja očekivanje log procjene da su podaci iz originalnog skupa podataka, a drugi pribrojnik predstavlja očekivanje procjene log vjerojatnosti da su podaci umjetno generirani.



Slika 2.1: Dolje-lijevo je prikazan D -dimenzionalni vektor (*D-dimensional noise vector*) koji predstavlja slučajni ulazni vektor za generator. Generator *Generator Network* je predstavljen crvenom bojom te u on prima opisani vektor. Generator generira nove (umjetne) slike *Fake Images* koje su ulaz u diskriminator *Discriminator Network* koji je predstavljen žutom bojom. Diskriminator još prima i originalan skup podataka *Real Images* i na izlazu daje predviženu oznaku koja je 0 ili 1 (ili je umjetna ili stvarna) *Predicted labels*. Preuzeto iz Sky

3. Kružni generativni suparnički modeli

Motivacija iza ovakvog modela proizlazi iz početne pretpostavke da za dvije domene postoje fundamentalne (često i puno više) sličnosti, primjerice između dvije slike istog prostora danju i noću jedina je razlika pozadinska svjetlost, između slike konja i zebre je osnovna razlika boja kože.

Sada možemo definirati i cilj kružnih generativnih suparničkih modela kao modela koji opaža neke specijalne karakteristike jedne i prilagođava ih (translatira) u drugu domenu što možemo grafički vidjeti na primjeru konja i zebre na slici 1.1

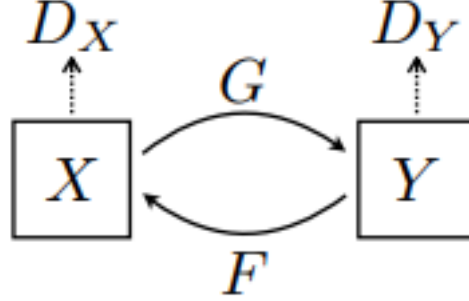
3.1. Matematički model

Kao što je opisano u uvodu ideja je npr. sliku iz jedne domene vjerno prikazati u drugoj domeni. Matematički zapisano želimo naučiti funkciju $G : X \rightarrow Y$ gdje je X prva domena, dok je Y ciljana domena, a funkcija G preslikava iz domene X u domenu Y , također želimo da se $G(X)$ ne razlikuje od domene Y tj. da diskriminator određuje da $G(X)$ pripada domeni Y sa što većom vjerojatnosti. Nadalje, ono što čini ovaj model kružnim je činjenica da želimo naučiti funkciju $F : Y \rightarrow X$ koja je inverzna funkciji $G : X \rightarrow Y$ i osigurava da $F(G(X)) \approx X$. i obrnuto. Sažeti cilj problema pregledno opisujemo jednadžbama (3.1)

$$\begin{aligned} G : X &\rightarrow Y, \\ F : Y &\rightarrow X \\ F(G(X)) &\approx X \\ G(F(Y)) &\approx Y \end{aligned} \tag{3.1}$$

G , F su bijekcije i međusobno su inverzni. G i F su generatori i svaki od njih ima pripadni diskriminator. Njihov odnos možemo vidjeti na slici 3.1. Na toj slici možemo

primijetiti da su upareni generator G i diskriminator D_Y te generator F i diskriminator D_X



Slika 3.1: Prikazan je osnovni princip rada. Generator G kao ulaz prima sliku iz domene X . Generira novu sliku koju predaje diskriminatoru D_Y . Generator F generira novu sliku iz domene Y i predaje je diskriminatoru D_X . Preuzeto iz Zhu et al. (2017)

3.1.1. Osnovne funkcije gubitka

Kao što je već opisano u 2 možemo definirati funkciju gubitka za par G i D_Y prikazano jednađžbom (3.2)

$$\mathcal{L}_{\mathcal{GAN}}(G, D_Y, X, Y) = \mathbf{E}_{y \sim p_{data}(y)}[\log D_Y(y)] + \mathbf{E}_{x \sim p_{data}(x)}[1 - \log D_Y(G(x))] \quad (3.2)$$

Analogno, definiramo i funkciju gubitka za par F i D_X prikazano jednađžbom (3.3)

$$\mathcal{L}_{\mathcal{GAN}}(F, D_X, Y, X) = \mathbf{E}_{x \sim p_{data}(x)}[\log D_X(x)] + \mathbf{E}_{y \sim p_{data}(y)}[1 - \log D_X(F(y))] \quad (3.3)$$

Nadalje, možemo definirati cilj funkcije $G : X \rightarrow Y$ prikazano jednađžbom (3.4)

$$G^*, D_Y^* = \min_G \max_{D_Y} \mathcal{L}_{\mathcal{GAN}}(G, D_Y, X, Y) \quad (3.4)$$

Analogno, definiramo cilj funkcije $F : Y \rightarrow X$ prikazano jednađžbom (3.5)

$$F^*, D_X^* = \min_F \max_{D_X} \mathcal{L}_{\mathcal{GAN}}(F, D_X, Y, X) \quad (3.5)$$

3.1.2. Kružno dosljedna funkcija gubitka

Opisanim funkcijama gubitka u poglavlju 3.1.1 je teoretski moguće (ako je duboki model velikog kapaciteta) mapirati slike iz domene X na neku slučajnu permutaciju slika

iz domene Y što svakako ne želimo. Naš cilj je translacijom iz domene X u domenu Y i natrag dobiti istu početnu sliku što možemo kao $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$ što nazivamo unaprijednom kružnom dosljednošću, analogno, definiramo unazadnu kružnu dosljednost kao $y \rightarrow F(y) \rightarrow G(F(y)) \rightarrow y$

Ove dva zahtjeva možemo ujediniti u jednoj funkciji gubitka prikazanoj u jednadžbi (3.6)

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbf{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbf{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1] \quad (3.6)$$

Ukupna funkcija gubitka je kombinacija tri funkcije gubitka prikazane jednadžbama (3.2), (3.3), (3.6) što je prikazano u jednadžbi (3.7) gdje parametar λ govori koliko je bitna kružna dosljednost u odnosu na osnovnu.

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\mathcal{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\mathcal{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F) \quad (3.7)$$

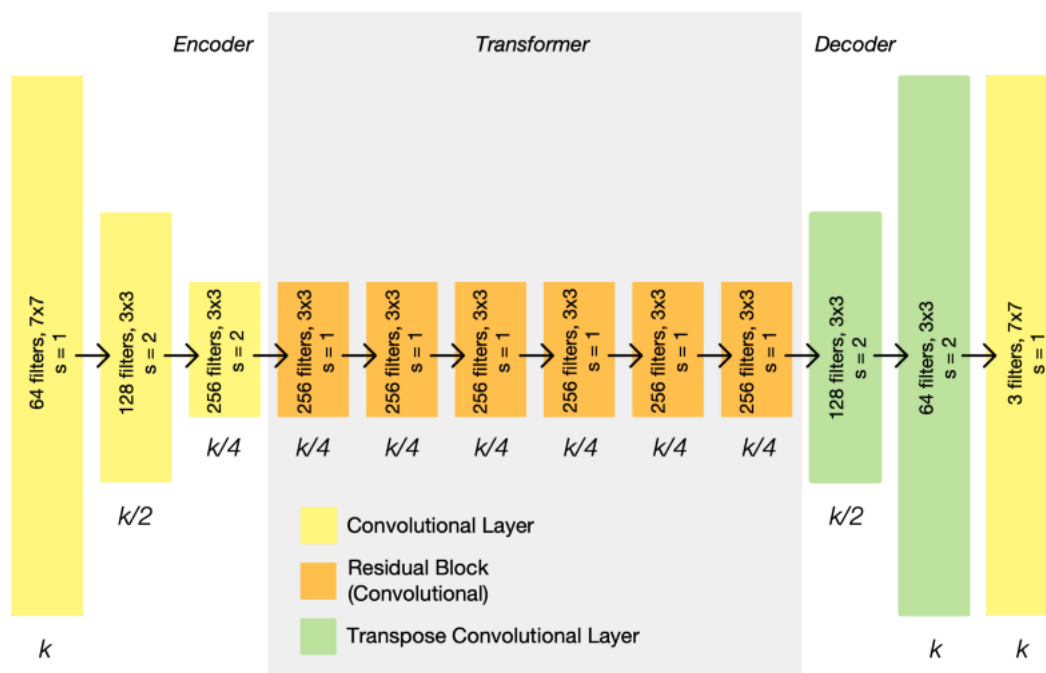
Konačno, ukupni cilj možemo zapisati pomoću jednadžbi (3.4), (3.5) i (3.7) kao (3.8)

$$G^*, F^* = \underset{G, F}{\operatorname{argmin}} \max_{D_X, D_Y} \mathcal{L}(G, F, D_X, D_Y) \quad (3.8)$$

4. Arhitektura

4.1. Generator

Arhitektura generatora je preuzeta iz Johnson et al. (2016), a možemo je zorno prikazati slikom 4.1. Iz slike možemo vidjeti tri glavna djela generatora: koder, transformator i dekodek.

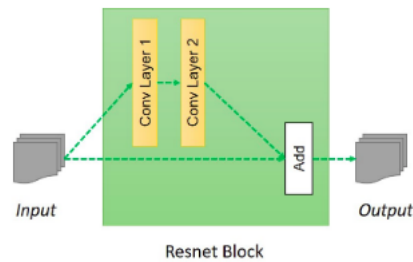


Slika 4.1: Arhitektura generatora, dodatno, nakon svakog sloja se nalazi normalizacije instance i leaky ReLU koji nisu prikazani na slici. Preuzeto iz Wolf (2018)

Uloga koder je da kodira, tj. izvuče bitne značajke iz ulazne slike. Koder se sastoji od 3 konvolucijska sloja od kojih svaki idući izvlači specifičnije značajke od prethodnog. Na izlazu koder dobivamo značajke ulazne slike generatora.

Uloga transformatora je kombinira susjedne značajke i određuje način transformacije značajki iz ulazne slike u izlaznu sliku. Transformator se sastoji od više rezidualnih blokova. Jedan rezidualni blok se sastoji od dva konvolucijska sloja koja su zbrojena s

ulazom i tako dobivamo da slika ostaje istih dimenzija kakva je bila na prije ulaska u blok. Rezidualni blok je prikazan na slici 4.2

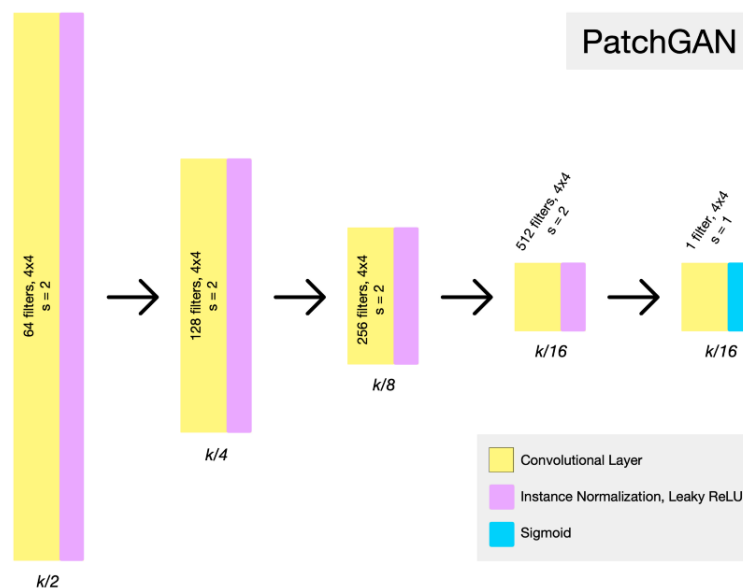


Slika 4.2: Prikaz rezidualnog bloka. Preuzeto iz Hardik Bansal

Uloga dekodera je suprotna ulozi koda. Na temelju transformiranih značajki ulazne slike se konstruira tražena izlazna slika. Dekoder se sastoji od dva sloja una-trajne konvolucije i jedan završni konvolucijski sloj kojim se dobiva slika istih dimenzija kao ulazna slika.

4.2. Diskriminator

Arhitektura diskriminatora se u potpunosti sastoji od konvolucijskih slojeva što je prikazano slikom 4.4. Ideja je se za pojedini dio slike (engl. *patch*) odredi je li stvaran ili umjetno generiran. Zato se i arhitektura još i zove 70×70 PatchGAN. 70×70 su dimenzije receptivnog polja zadnjeg konvolucijskog sloja.



Slika 4.3: Arhitektura diskriminatora. Preuzeta iz Wolf (2018)

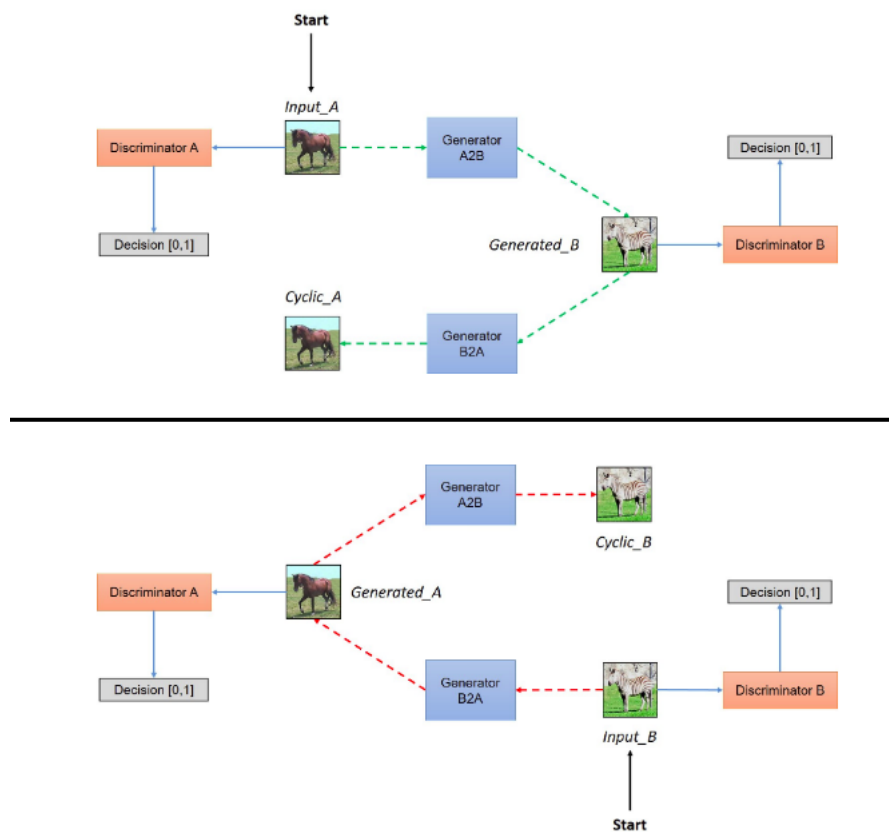
4.3. Treniranje

U procesu treniranja je umjetno negativne log izglednosti korištena metoda srednje kvadratne pogreške koja je stabilnija i generira slike visoke kvalitete. Generator minimizira izraz (4.1), dok diskriminator minimizira izraz (4.2)

$$\mathbf{E}_{x \sim p_{data}(x)}[(D(G(X)) - 1)^2] \quad (4.1)$$

$$\mathbf{E}_{x \sim p_{data}(x)}[D(G(X))^2] + \mathbf{E}_{y \sim p_{data}(y)}[D(Y) - 1] \quad (4.2)$$

U odnosu na klasično ažuriranje generatora kod GAN-a gdje se koristi zadnja generirana slika, autori predlažu ažuriranje na temelju zadnjih 50 generiranih slika. U jednadžbi (3.7) je odabrana $\lambda = 10$ kako bi se dala prednost kružnoj funkciji gubitka u odnosu na klasične GAN-ove funkcije gubitka dane jednadžbama (3.2) i (3.3)

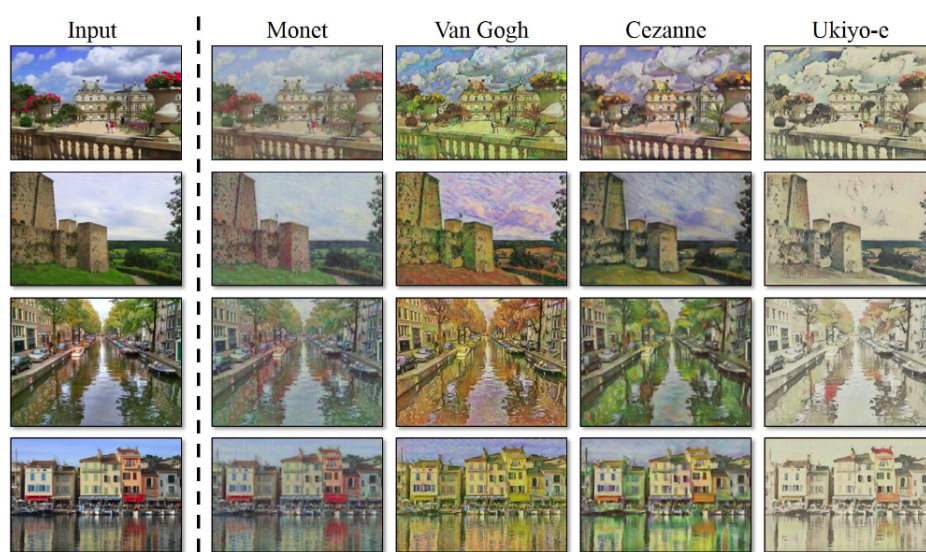


Slika 4.4: Na gornjoj slici vidimo da se iz slika iz početne domene X šalje u generator $A2B$ koji generira novu sliku sličniju domeni Y u svakom koraku. Ta slika se prosljeđuje *diskriminatoru B*, a originalna slika konja iz domene X se prosljeđuje *diskriminatoru A*. Generirana slika iz domene se šalje *generatoru B2A* koji bi trebao generirati identičnu sliku iz domene X. Analogan je postupak na donjoj slici kada počinjemo od slike iz domene Y

5. Najvažniji rezultati i primjeri

U Zhu et al. (2017) su predstavljeni najvažniji rezultati. Dana je usporedba sa sličnim idejama nenadziranog učenja koje koriste ideju GAN-a te je pokazano da je kružni GAN trenutno najbolji, iako, u odnosu na metode koje se baziraju na nadziranom učenju i dalje je lošiji. Napravljena je studija slučaja, te su pokazani rezultati bez korištenja svake od funkcija gubitaka i pokazano je da su vrlo bitne u dobivanju boljih rezultata.

Primjer zadatka u kojem se fotografija pretvara u stilove različitih likovnih umjetnika je prikazana u 5.1



Slika 5.1: Pretvorba iz fotografije u različite stilove slikara: Monet, Van Gogh, Cezanne, Ukiyo-e Preuzeto iz Zhu et al. (2017)

Jedan od zadataka na kojem su evaluirani je transformacija objekta npr. konju u zebro što je već prikazano na slici 1.1. Evaluacija je izvršena i na zadatku pretvaranja slike u zimskom dobu u ljetno doba i obrnuto što je prikazano na slici 5.2.

Također, prikazani su i neki slučajevi u kojima je loše translirana slika iz jedne domene u drugu, npr. vidimo na primjeru slike 5.3 gdje se osoba obojila u boje zebre, a razlog je što su u skupu za treniranje bile samo slike konja/zebri bez ljudi.



Slika 5.2: Pretvorba slike zimskog doba u ljetno i obrnuto. Preuzeto iz Zhu et al. (2017)



Slika 5.3: Model je obojio i osobu. Preuzeto iz Zhu et al. (2017)

6. Zaključak

Na kraju seminarskog rada možemo zaključiti da je ova metoda bitni pomak u zadacima translacije slike iz jedne domene u drugu, ali koristeći nenadzirano učenje u odnosu na ostale nenadzirane metode. Prednost nenadziranih metoda je što ne moraju imati uparene slike iz različitih metoda. Dakako, ovaj model nije pogodan za zadatke poput pretvaranja mačke u psa i obrnuto, ali je ostavljeno za budući rad takva problematika.

7. Literatura

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, i Yoshua Bengio. Generative adversarial networks, 2014.

Archit Rathore Hardik Bansal. *CycleGAN: Learning to Translate Images (Without Paired Training Data)*. URL <https://hardikbansal.github.io/CycleGANBlog/>.

Justin Johnson, Alexandre Alahi, i Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution, 2016.

A *Beginner's Guide to Generative Adversarial Networks (GANs)*. Skymind. URL <https://skymind.ai/wiki/generative-adversarial-network-gan>.

Sara Wolf. *CycleGAN: Learning to Translate Images (Without Paired Training Data)*, 2018. URL <https://towardsdatascience.com/cyclegan-learning-to-translate-images-without-paired-training-data->

Jun-Yan Zhu, Taesung Park, Phillip Isola, i Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. U *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

8. Sažetak

U seminarskom radu je opisan rad Zhu et al. (2017) kao inovativno rješenje koje translatira slike iz jedne domene u drugu, ali slika iz jedne domene ne mora imati uparenu sliku iz druge domene. Predstavljena je motivacija, matematički model, arhitektura modela i način treniranja. Dani su primjeri rezultata, zadataka i odgovarajući zaključak. Zainteresiranog čitatelja svakako upućujemo na stranicu <https://junyanz.github.io/CycleGAN/> gdje može pronaći još mnoštvo zanimljivih informacija o kružnim generativnim suparničkim modelima.