Univerzitet u Beogradu - Elektrotehnički fakultet

Katedra za signale i sisteme

DIPLOMSKI RAD

Konvolucione neuralne mreže za transfer stila

Kandidat
Nikola Dimić, br. Indeksa 2020/0128

Mentor
dr Predrag Tadić, vanredni profesor

Beograd, jul 2024. godine

SADRŽAJ

1 UVOD

Neuralni prenos stila (eng. *Neural Style Transfer*) je pojam koji se odnosi na klasu algoritama koji manipulišu digitalnim slikama ili video zapisima kako bi usvojili izgled ili vizuelni stil druge slike ili video zapisa. NST (*Neural Style Transfer*) algoritme karakteriše upotreba dubokih neuralnih mreža (eng. *Deep Neural Networks*) radi transformacije slike. Uobičajeni primer upotrebe NST-a je stvaranje veštačkih umetničkih dela od fotografija prenošenjem izgleda slika poznatih umetnika na fotografije prikupljene od strane korisnika. Nekoliko značajnih mobilnih aplikacija koristi NST tehnike u tu svrhu, uključujući DeepArt i Prisma. Ovu metodu su koristili umetnici i dizajneri širom sveta za razvoj novih umetničkih dela zasnovanih na postojećim stilovima.

NST je primer stilizacije slike, problem koji se proučava više od dve decenije u oblasti nefotorealističnog renderovanja. Prva dva algoritma za prenos stila koji su učili na osnovu primera bili su analogije slika (eng. *Image Analogies*) [1] i prošivanje slika (eng. *Image Quilting*) [2]. Obe ove metode su zasnovane na algoritmima za sintezu tekstura. Imajući na raspolaganju fotografiju napravljenu od strane korisnika i umetničko delo, transformacija bi se mogla naučiti, a zatim primeniti na kreiranje novog umetničkog dela od nove fotografije, po analogiji. Ukoliko nije bila dostupna nijedna fotografija za obuku, ona bi trebalo da se napravi obradom ulaznog umetničkog dela; prošivanje slika nije zahtevalo ovaj korak obrade, iako je demonstrirano samo na jednom stilu.

NST se po prvi put pojavljuje u radu [3] koji je prihvaćen na konferenciji CVPR (*Conference on Computer Vision and Pattern Recognition*) 2016. godine. Originalni rad je koristio arhitekturu VGG-19 koja je prethodno obučena za prepoznavanje objekata koristeći *ImageNet* skup podataka. Početkom 2017. godine *Google AI* je predstavio metod koji omogućava jednoj dubokoj konvolucionoj mreži za prenos stilova da nauči više stilova u isto vreme [4]. Ovaj algoritam dozvoljava interpolaciju stila u realnom vremenu, čak i kada se radi na video snimcima.

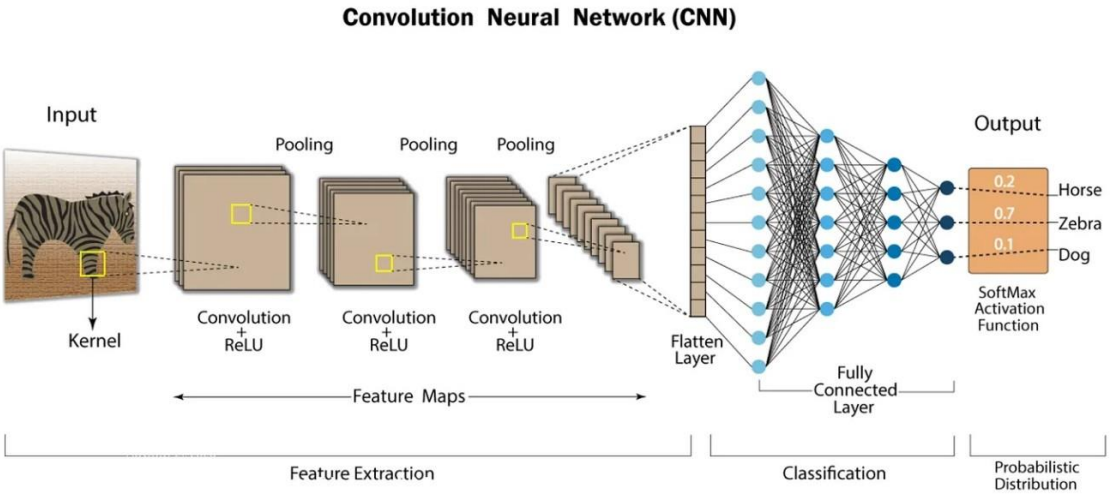
Primena NST-a obuhvata različite oblasti, uključujući umetnost, dizajn, film i video igre. U umetnosti i dizajnu NST omogućava umetnicima da eksperimentišu sa različitim stilovima i tehnikama, stvarajući jedinstvena umetnička dela koja kombinuju elemente klasičnih slikarskih stilova sa savremenim motivima. U industriji filma i video igara NST se koristi za generisanje impresivnih vizuelnih efekata i stilizovanje scena, dodajući vizuelnu dinamiku i kreativnost koja je ranije bila teško ostvariva. NST takođe nalazi primenu u marketingu gde omogućava stvaranje privlačnih vizuala i prilagođavanje reklamnih kampanja različitim estetskim preferencijama ciljne publike. Osim toga, NST se koristi u edukaciji, gde pomaže studentima da bolje razumeju umetničke stilove i tehnike kroz interaktivne i praktične primere. Ova tehnika takođe ima potencijal u medicini, gde može pomoći u vizualizaciji podataka na način koji olakšava interpretaciju i analizu. Kroz sve ove oblasti, NST pokazuje kako napredak u veštačkoj inteligenciji može obogatiti kreativne procese i proširiti granice tradicionalnih metoda stvaranja i analize vizuelnih sadržaja.

Cilj ovog rada je da se na jednom mestu pomenu i objasne svi koncepti neophodni za razumevanje funkcionisanja *Neural Style Transfer* algoritama ali i prikažu unapređenja proistekla iz kasnijih radova vezana za performanse i brzinu treniranja ovih mreža. U nastavku su prikazane metode koje se koriste za realizaciju jednog sistema za *Neural Style Transfer*, a kasnije i rezultati dobijeni primenom izloženih metoda.

2 METODOLOGIJA RADA

2.1 Konvolucione mreže

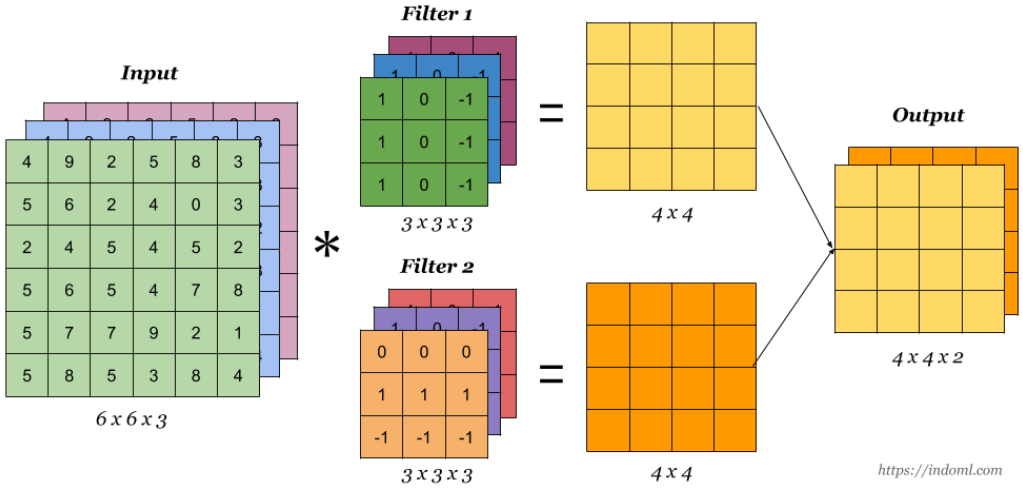
Konvolucione neuralne mreže (eng. *Convolutional Neural Networks*) su tip *feedforward* neuralnih mreža koje se pojavljuju kao rešenja velikog broja problema kompjuterske vizije (eng *Computer Vision*). Kod ovih mreža nije neophodno ručno pronalaženje adekvatnih obeležja, već se njihova ekstrakcija vrši unutar same mreže, uz pomoć niza filtara koji imaju svoju funkciju. Primer konvolucione mreže prikazan je na slici 1. Za razliku od tipičnih *feedforward* mreža, konvolucione mreže su izrazito pogodne za klasifikaciju slika, jer uzimaju u obzir i prostorni raspored piksela slike, a ne samo njihove intenzitete. Tipična konvoluciona mreža se sastoji iz kombinacije konvolucionih, *max pooling* i ReLU slojeva, zaključno sa nizom potpuno povezanih slojeva i *softmax* slojem na kraju [5]. Izlaz svakog sledećeg bloka (pod blokom se podrazumeva sekvenca konvolucionog, *max pooling* sloja i aktivacione funkcije) mreže nosi sve kompleksniju informaciju o slici, tako da kasniji slojevi mogu prepoznati i veoma komplikovane strukture (poput pojedinih delova tela ili drugih karakterističnih oblika). Mogućnost mreže da nauči komplikovane reprezentacije ulazne slike čini je veoma moćnim alatom za rešavanje niza problema vezanih za detekciju sadržaja slike. U nastavku je detaljno objašnjena uloga pojedinih delova mreže.



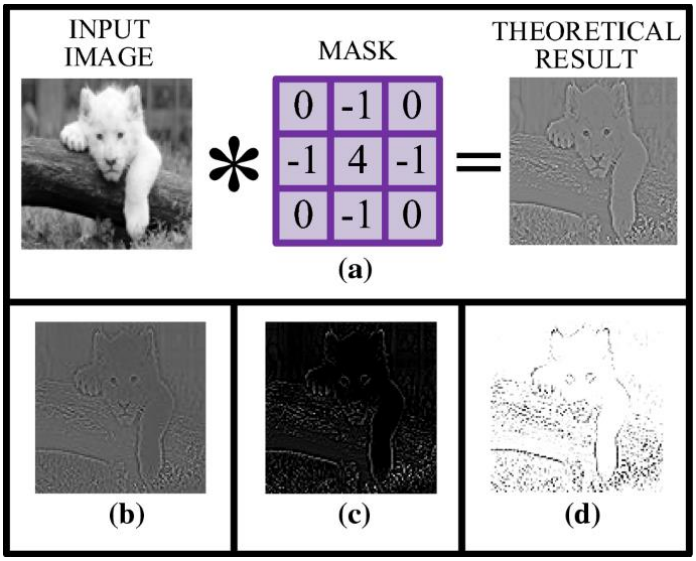
Slika 1. Prikaz karakteristične konvolucione mreže, preuzeto sa sajta https://www.linkedin.com/

2.1.1 Konvolucioni sloj

Konvolucioni sloj je ključna komponenta konvolucionih neuralnih mreža. Ovaj sloj poseduje specifične funkcionalnosti koje omogućavaju mreži da automatski i efikasno uči prostorne hijerarhije na osnovu ulaznih podataka. Glavna komponenta svakog konvolucionog sloja su filtri. Filtri su male matrice, tipično dimenzija 3×3, 5×5 ili 7×7 koje 'klize' preko ulazne slike i omogućavaju učenje različitih karakteristika slike poput tekstura, ivica, uglova itd. Proces prolaska ovakvih filtara preko slike i množenja odgovarajućih elemenata ulazne matrice sa elementima filtra i sabiranje rezultata naziva se konvolucija. Rezultat konvolucije ulazne matrice i filtra je matrica istih ili izmenjenih dimenzija koja predstavlja odziv ulazne matrice na dati filtar. Na slikama 2 i 3 su prikazani konvolucioni sloj i primer filtra:



Slika 2. Rezultat konvolucije ulazne slike i filtra, preuzeto sa https://indoml.com/

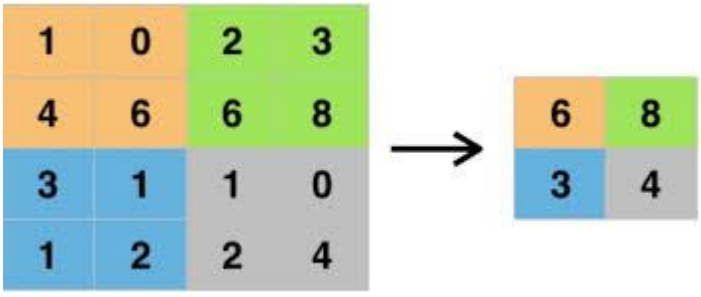


Slika 3. Primer filtra koji se koristi za izdvajanje ivica, preuzeto sa https://link.springer.com/, pod a je prikazana Laplasijan maska, pod b konvolucija ulazne slike sa ovom maskom , a pod c i d slike koje su binarizovane na osnovu odziva pod b

Sa ciljem da očuvamo dimenzije ulazne slike i na izlazu konvolucionog sloja koristi se tehnika po imenu *padding*. Ovo je proces proširivanja slike dodatnim pikselima po ivici. Postoji nekoliko načina da se izvrši ovo proširivanje poput ponavljanja ili preslikavanja već postojećih ivičnih piksela i u većini slučajeva sve ove metode daju zadovoljavajuće rezultate. Rezultat primene većeg broja filtara na ulaznu sliku je niz matrica koje predstavljaju odzive slike na sve filtre i često se nazivaju mape obeležja (eng. *feature maps*). Česta je praksa da se izlaz konvolucionog sloja dovede na ReLU aktivacionu funkciju koja u kompletnu mrežu unosi nelinearnost i poboljšava kapacitet učenja samog modela.

2.1.2 Max pooling sloj

Još jedan tip sloja koji se koristi unutar konvolucionih mreža je *max pooling* sloj. Kroz formirane mape obeležja se prolazi maskom koja je najčešće dimenzija 2×2, 3×3 ili 4×4 sa pomerajem koji je jednak dimenziji maske. Na primer, ukoliko je maska dimenzija 2×2 prolazimo kroz sve podmatrice ulazne matrice koje su dimenzija 2×2 i za svaku od njih nalazimo maksimum. Ovime postižemo nekoliko stvari. Prvo, smanjujemo dimenzionalnost problema. Ukoliko bi dimenzije ulazne slike ostale nepromenjene pri prolasku kroz svaki sloj količina operacija koje se izvršavaju tokom treninga bila bi prevelika. Drugo, uzimanjem maksimuma svake podmatrice zadržavamo najbitniju informaciju i sprečavamo preobučavanje mreže usled prevelikog broja parametara. Zaključno, *max pooling* slojevi doprinose robusnosti modela s obziroma da čine mrežu otpornom na male pomeraje i varijacije na ulazu. Još jedan parametar koji figuriše u strukturi *max pooling* sloja je pomeraj (eng. *stride*). Pomerajem je određeno za koliko se pomera maska prilikom prolaska preko ulazne matrice. Na primer, ukoliko je $stride = 2$ tada će se dimenzije ulazne matrice prilikom prolaska kroz *max pooling* sloj smanjiti duplo. Na slici 4 je ilustrovano funkcionisanje *max pooling* sloja:

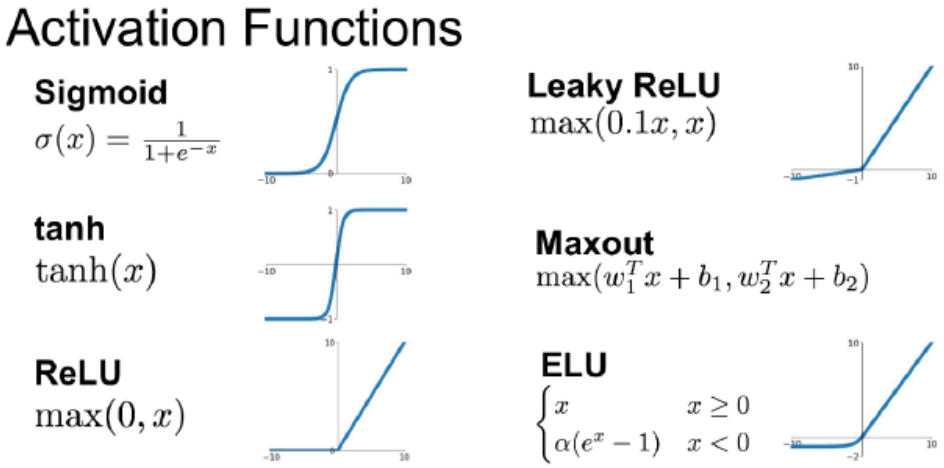


Slika 4: Ilustracija rada max pooling sloja, preuzeto sa https://medium.com/

Važno je napomenuti da se umesto operacije nalaženja maksimuma može raditi i sa srednjom vrednošću, što je pristup koji se koristi u pojedinim aplikacijama. Ipak, u najvećem broju slučajeva najbolje performanse se dobijaju korišćenjem maksimuma.

2.1.3 Aktivacione funkcije

Aktivacione funkcije igraju značajnu ulogu u neuralnim mrežama s obzirom na to da omogućavaju mreži da nauči složene nelinearne odnose između ulaznih i izlaznih podataka. U konvolucionim neuralnim mrežama u većini slučajeva se koristi ReLU aktivaciona funkcija, jer ona rešava problem nestajućih gradijenata i omogućava realizaciju veoma dubokih neuralnih mreža. U manjem broju slučajeva koriste se i varijante ReLU funkcije poput *leaky* ReLU ali i druge funkcije poput *sigmoid* i *tanh* funkcija za specifične namene. Na slici 5 prikazane su aktivacione funkcije koje se najčešće koriste:



Slika 5. Prikaz različitih aktivacionih funkcija, preuzeto sa http://www.matf.bg.ac.rs/

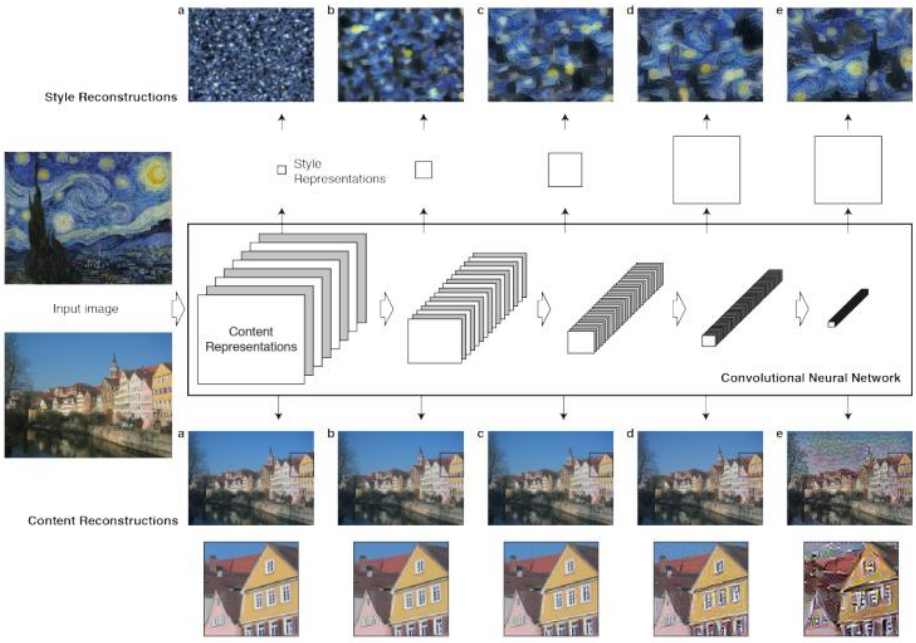
2.1.4 Flatten, potpuno povezani i softmax slojevi

U najvećem broju slučajeva, nakon što serija konvolucionih slojeva izvrši svojevrsnu ekstrakciju obeležja, ulaz u drugi deo mreže ima dimenzije m×n×f, gde je f broj filtara poslednjeg konvolucionog sloja. Za potrebe klasifikacije potrebno je ovaj ulaz 'razvući' u vektor kako bi on bio doveden na ulaz standardne potpuno povezane (eng. *fully connected)* neuralne mreže. U te svrhe koristimo *flatten* sloj, koji će pomenuti ulaz preoblikovati u vektor dužine m×n×f. Ovakav vektor dovodi se na niz potpuno povezanih slojeva završno sa *softmax* slojem na kraju, ukoliko je u pitanju problem klasifikacije. Ovaj deo mreže pomenut je radi kompletnosti, međutim, on nema značaja kada se govori o algoritmima za transfer stila. U nastavku će biti objašnjeno kako se unapred istrenirana konvoluciona neuralna mreža može upotrebiti za realizaciju ovih algoritama.

2.2 Realizacija prenosa stila optimizacijom

2.2.1 Reprezentacije sadržaja slike

Pokazuje se da konvolucione mreže poseduju svojstva koja su veoma pogodna pri rešavanju problema transfera stila. Naime, prilikom obučavanja za prepoznavanje objekata, ove mreže formiraju reprezentacije slike koje čine informaciju o objektima sve više eksplicitnom, idući kroz hijerarhiju obrade. Imajući to u vidu, ulazna slika se prolazeći kroz slojeve konvolucione mreže tranformiše u reprezentacije koje sve više brinu o stvarnom sadržaju slike, a sve manje o intenzitetima pojedinačnih piksela [3]. Ove reprezentacije je moguće i vizualizovati što je prikazano na slici 6:



Slika 6. Reprezentacije ulazne slike, preuzeto iz rada [3]

Ukoliko se vratimo na početni problem, cilj nam je da generišemo sliku takvu da njen sadržaj liči na ulaznu sliku sadržaja (eng. *style image*), a njen stil liči na ulaznu sliku stila (eng. *style image*). Ovakva postavka problema nije u potpunosti jasna, jer je potrebno definisati pojam sadržaja i pojam stila. U jednom od radova [4] pominje se ovakva definicija:

- Dve slike imaju sličan sadržaj ukoliko su njihova obeležja visokog nivoa (eng. *high-level*), izvučena od strane prethodno istreniranog klasifikatora slična po Euklidskoj distanci

- Dve slike imaju sličan stil ukoliko njihova obeležja visokog nivoa izvučena od strane prethodno istreniranog klasifikatora imaju sličnu statistiku, konkretno, ukoliko razlika njihovih *Gram*-ovih matrica ima malu *Frobenius*-ovu normu

Generalno, svaki sloj u mreži definiše nelinearan skup filtara čija složenost raste sa pozicijom sloja u mreži. Otuda je data ulazna slika $\vec{x}$ kodirana u svakom sloju konvolucione mreže odzivima filtara na tu sliku. Sloj sa N_l različitih filtera ima N_l mapa obeležja veličine M_l, gde je M_l je visina puta širina mape obeležja. Dakle, odzivi u sloju l se mogu čuvati u matrici $F^l \in R^{N_l * M_l}$ gde je F_{ij}^l aktivacija i-tog filtra na poziciji j u sloju l.

Sa ciljem da vizualizujemo reprezentacije pojedinih slojeva mogli bismo na ulaz mreže da dovedemo beli Gausov šum i da sprovedemo *backpropagation* algoritam kako bismo pronašli sliku čija se reprezentacija istog sloja slaže sa reprezentacijom početne slike. Uzmimo da su, redom, $\vec{p}$ i $\vec{x}$ originalna i generisana slika a P^l i F^l njihove reprezentacije sloja l. Kvadratna greška reprezentacije generisane slike na osnovu l-tog sloja ima oblik:

$$L_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2}\sum_{i,j}(F_{ij}^l - P_{ij}^l)^2$$

Izvod ovog izraza po aktivacijama koje potiču od generisane slike je:

$$\frac{\partial L_{content}}{\partial F_{ij}^l} = (F_l - P_l)_{ij} \text{ za } F_{ij}^l > 0$$

$$\frac{\partial L_{content}}{\partial F_{ij}^l} = 0 \text{ za } F_{ij}^l < 0$$

Koristeći *backpropagation* algoritam na standardan način moguće je pronaći izvod funkcije greške po svim pikselima generisane slike $\vec{x}$ i na taj način menjati sliku tako da njen sadržaj što više liči na sadržaj slike $\vec{p}$.

2.2.2 Reprezentacije stila slike

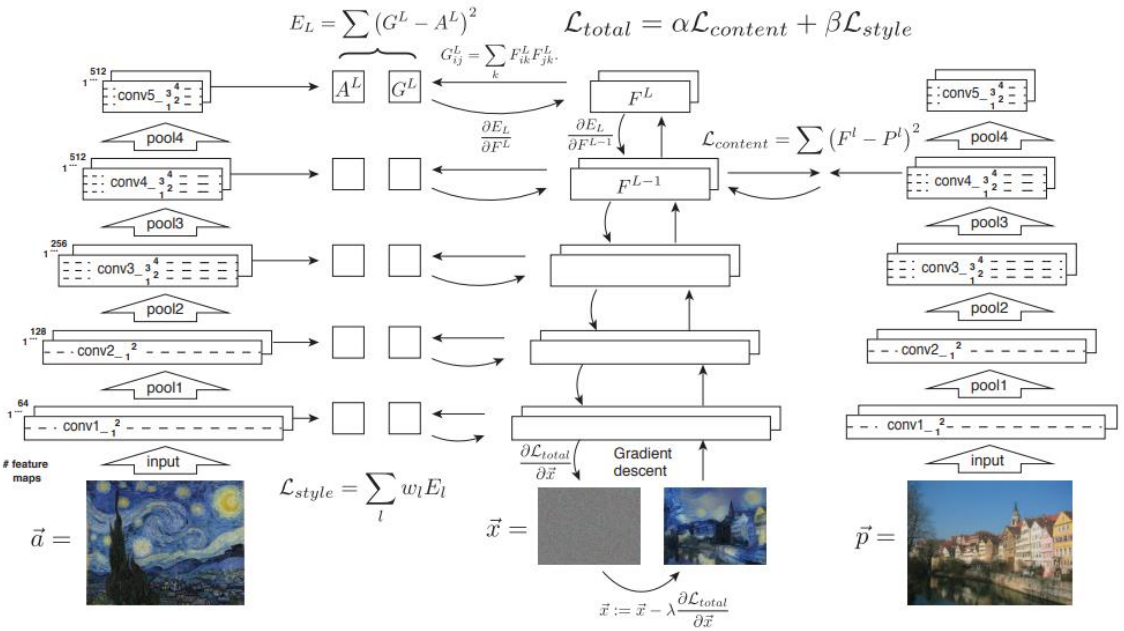
Sa ciljem da se prikaže reprezentacija stila ulazne slike, koristi se prostor karakteristika dizajniran da uhvati informacije o teksturi. Ovaj prostor obeležja može se izgraditi na osnovu odziva filtara u bilo kom sloju mreže. Obeležja se zasnivaju na korelaciji između parova odziva na različite filtre i čuvaju se u takozvanoj *Gram*-ovoj matrici $G^l \in R^{N_l \times N_l}$, gde je G_{ij}^l unutrašnji proizvod vektorizovanih mapa obeležja i i j.

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l$$

Intuitivno *Gram*-ova matrica govori o tome koji se odzivi aktiviraju istovremeno, te na taj način nosi svojevrsnu informaciju o teksturi pa i stilu same slike. U nastavku, smatraće se da dve slike imaju sličan stil ukoliko je razlika njihovih *Gram*-ovih matrica izračunatih za slojeve koji su od interesa što manja. Kao i malopre, može se formirati funkcija gubitka (eng. *loss function*) čijom minimizacijom dobijamo sliku sa stilom koji veoma liči na zadati.

2.2.3 Style transfer algoritam

U ovom delu je prikazana je verzija algoritma koja se pojavljuje u radovima [3] i [6]. Pretpostavimo da je sa $\vec{a}$ označena slika ciljnog stila, sa $\vec{p}$ označena slika ciljnog sadržaja, a sa $\vec{x}$ generisana slika. Na slici 7 ilustrovan je rad algoritma:



Slika 7. NST algoritam, preuzeto iz rada [6]

Kao što je već rečeno, cilj je da generisana slika $\vec{x}$ istovremeno poseduje sadržaj slike $\vec{p}$ i stil slike $\vec{a}$. Na početku, $\vec{x}$ inicijalizujemo belim šumom. Slika $\vec{a}$ se propagira kroz mrežu i računaju se i čuvaju reprezentacije stila označene sa A^l. Istovremeno se kroz mrežu propagiraju slike $\vec{p}$ i $\vec{x}$ pri čemu se za sliku $\vec{p}$ računaju reprezentacije sadržaja, a za generisanu sliku $\vec{x}$ reprezentacije sadržaja i stila označene redom sa F^l i G^l. Za svaki sloj koji učestvuje u reprezentaciji stila, računa se suma kvadrata razlika elemenata matrica G^l i A^l i ona predstavlja grešku stila označenu sa L_{style}. Takođe, grešku sadržaja $L_{content}$ formiramo kao srednju kvadratnu grešku matrice F^l u odnosu na P^l. Konačno, funkcija gubitka L_{total} predstavlja linearnu kombinaciju grešaka stila i sadržaja. Moguće je izračunati njen izvod po svim pikselima ulazne slike $\vec{x}$. Gradijenti se koriste kako bi se iterativnim postupkom ažurirala ulazna slika sve do trenutka kada ona istovremeno oponaša sadržaj slike $\vec{p}$ i stil slike $\vec{a}$.

Doprinos sloja l ukupnoj greški stila ima oblik:

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2$$

a funkcija gubitka stila je:

$$L_{style}(\vec{a},\vec{x}) = \sum_{i=0}^{L} w_l E_l$$

gde su w_l težine pripisane doprinosima različitih slojeva funkciji gubitka. Takođe, izvode grešaka E_l po aktivacijama iz l-tog sloja moguće je izračunati:

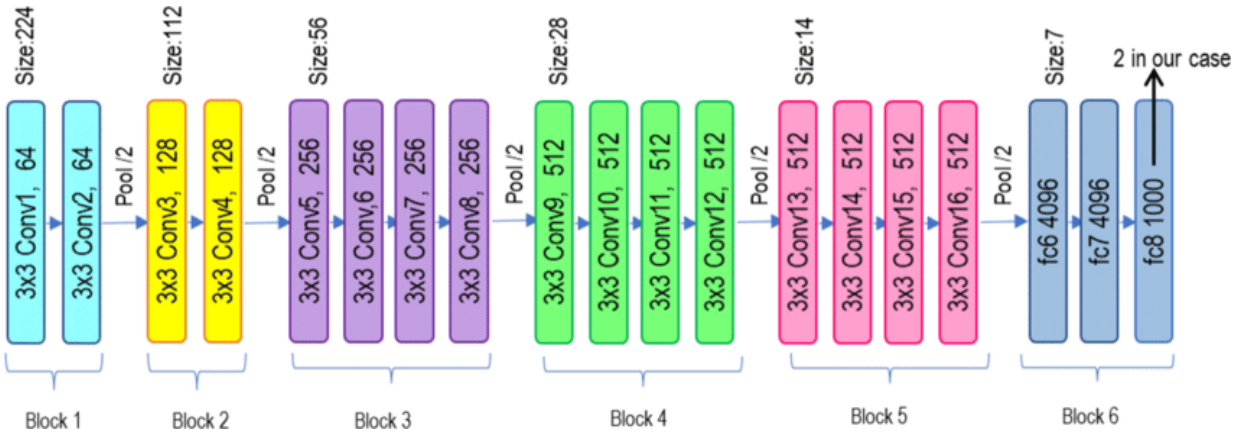
$$\frac{\partial E_l}{\partial F_{ij}^l} = \begin{cases} \frac{1}{4N_l^2 M_l^2}((F^l)^T(G^l - A^l))_{ji}, za\ F_{ij}^l > 0 \\ 0,\ \ u\ suprotnom \end{cases}$$

Dalje, gradijenti u odnosu na piksele ulazne slike $\vec{x}$ mogu se izračunati primenom *backpropagation* algoritma. Konačna funkcija gubitka ima oblik:

$$L_{total}(\vec{p},\vec{a},\vec{x}) = \alpha L_{content}(\vec{p},\vec{x}) + \beta L_{style}(\vec{a},\vec{x})$$

gde su α i β težine koje pripisujemo greškama sadržaja i stila i čijim se podešavanjem može birati koja od ove dve greške je bitnija prilikom optimizacije. Ovo se često radi tako što se jedan od ova dva parametra fiksira, a drugi menja, jer je jedino odnos ovih parametara od interesa.

Važno je naglasiti da je ovo jedan od prvih algoritama ovog tipa koji se pojavio i da on itekako nije savršen. Takođe, arhitektura konvolucione mreže koja se koristi za ekstrakciju obeležja u velikoj meri utiče na rezultate. U najvećem broju primena NST algoritma koristi se VGG-19 arhitektura konvolucione mreže sa 16 konvolucionih i 5 *pooling* slojeva. Ova arhitektura je prikazana na slici 8:

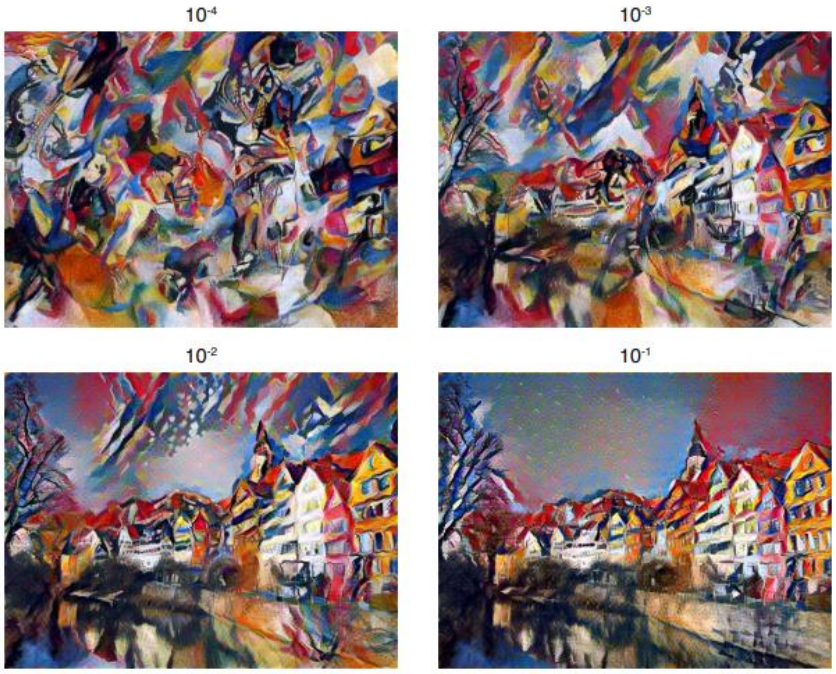


Slika 8. Arhitektura VGG-19 konvolucione mreže, preuzeto sa https://www.researchgate.net/

Takođe, treba primetiti da je izbor slojeva koji se koriste za formiranje funkcije gubitka od presudnog značaja. Rezultati koji se nalaze na slici 9 sintetizovani su korišćenjem funkcije gubitka sadržaja formirane na sloju „conv4_2“ i funkcije gubitka stila formirane na slojevima „*conv1_1*“, „*conv2_1*“, „*conv3_1*“, „*conv4_1*“ i „*conv5_1*“ (oznake preuzete sa slike 7).

Slika 9. Rezultati dobijeni primenom NST algoritma, preuzeto iz rada [6]

Kod dobijenih slika korićene su različite vrednosti odnosa α/β. Ovaj odnos uzima vrednosti 1×10^{-3} (slika B), 8×10^{-4} (slika C), 5×10^{-3} (slika D), 5×10^{-4} (slike E i F). Na slici 10 prikazan je i uticaj odnosa α/β na dobijene rezultate:



Slika 10. Uticaj težina a i b na dobijene rezulate, preuzeto iz rada [6]

Primećuje se da se za male vrednosti ovog odnosa dobijaju slike koje skoro u potpunosti zanemaruju zadatu sliku sadržaja, ali se stil u potpunosti čuva. Sa druge strane, ukoliko je ovaj odnos veći, detalji zadate slike sadržaja postaju uočljiviji. Jasno je da optimalan odnos α/β zavisi od više faktora i neophodno je naći adekvatnu vrednost kako bi se dobili zadovoljavajući rezultati.

2.3 Realizacija prenosa stila primenom feedforward mreže

2.3.1 Ideja

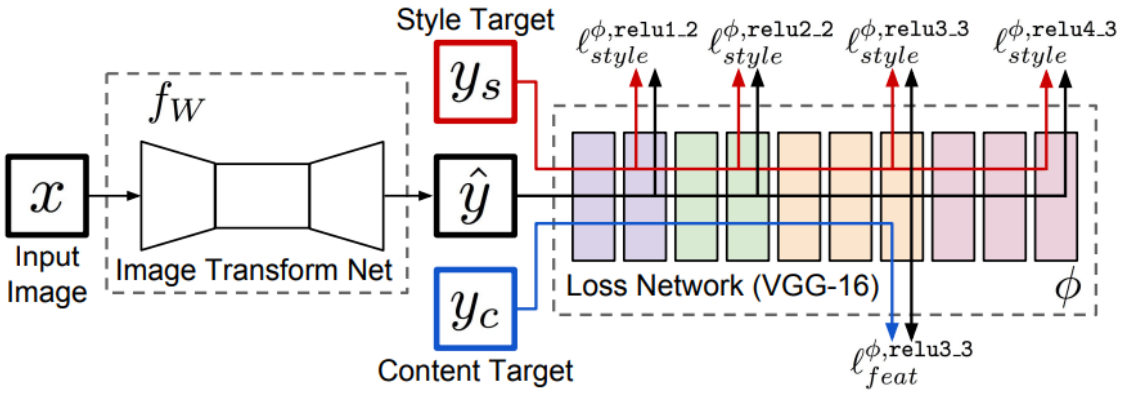
Navedeni pristup, iako daje veoma dobre rezultate, ima jednu veliku manu. Generisanje nove slike NST algoritmom za zadate slike sadržaja i stila zahteva rešavanje optimizacionog problema što može biti veoma skup i spor proces. Veliko trajanje generisanja slike čini ovaj postupak praktično neprimenljivim u *real-time* aplikacijama i značajno sužava njegovu primenu.

Veliki broj klasičnih problema mogu se formulisati kao problemi transformacije slike [7], gde sistem prima ulaznu sliku, a kao rezultat vraća izlaznu. Neki primeri koji se pojavljuju u obradi slike su super-rezolucija, kolorizacija i odšumljavanje, kod kojih je ulaz degradirana slika, a izlaz slika sa poboljšanim kvalitetom. Jedan od načina za rešavanje ovakvih problema je treniranje *feedforward* konvolucione neuralne mreže supervizijski koristeći funkciju gubitka koja se formira od razlika pojedinačnih piksela dobijenih i referentnih slika. Međutim, ovako formirana funkcija gubitka ne uzima u obzir perceptualnu informaciju koju nose slike. Naime, ukoliko posmatramo dve slike, pri čemu je druga dobijena od prve translacijom njenih piksela

za jedno mesto udesno, ovakva funkcija gubitka može imati veliku vrednost iako je informacija na obe slike praktično ista.

2.3.2 Arhitektura

U radu [7] predložena je arhitektura koja zadržava kvalitet rezultata iz radova [3] i [6], ali unosi ogromno poboljšanje koje se tiče vremena izvršavanja. Ova arhitektura je prikazana na slici 11:



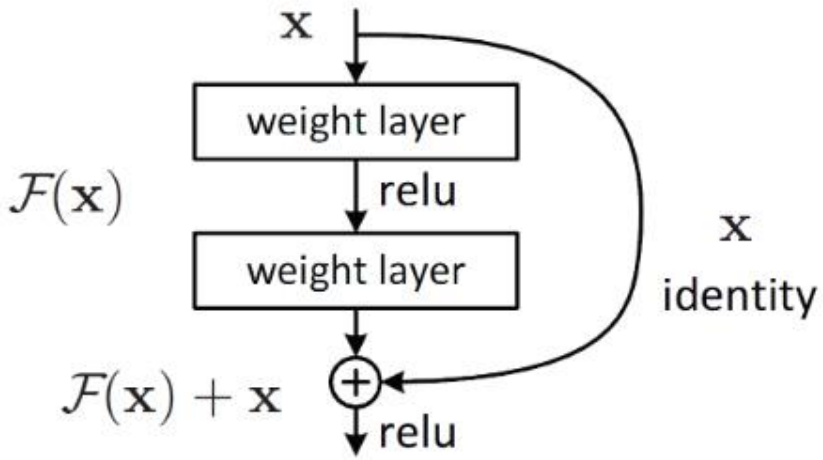
Slika 11. Arhitektura za NST koja koristi feedforward mrežu, preuzeto iz rada [7]

Sistem se sastoji iz dve ključne komponente: transformacione mreže (eng. *transformation network*) f_w i mreže greške (eng. *loss network*) ϕ pomoću koje su definisane funkcije gubitka $l_1, l_2, \dots, l_k$. Transformaciona mreža je duboka rezidualna konvoluciona neuralna mreža koja se može parametrizovati matricom težina W i koja transformiše ulazne slike x u izlazne slike $\hat{y}$ preslikavanjem $\hat{y} = f_w(x)$. Ova mreža poseduje sledeća svojstva:

- Ne koriste se *pooling* slojevi, umesto njih se za *upsampling* i *downsampling* unutar mreže koriste pomerene i delimično pomerene konvolucije (eng. *strided/partialy strided convolutions*)
- 5 rezidualnih blokova
- Nakon svih konvolucionih slojeva sa izuzetkom izlaznog sloja nalaze se prostorna *batch* normalizacija i ReLU nelinearnost
- Izlazni sloj koristi skaliranu *tanh* funkciju kako bi osigurao da izlazna slika ima vrednosti piksela u opsegu [0,255]
- Prvi i poslednji konvolucioni sloj koriste kernele veličine 9×9
- Svi ostali slojevi koriste kernele veličine 3×3
- Ulaz i izlaz ove mreže moraju imati dimenzije $3 \times 256 \times 256$

Iako na prvi pogled ista mreža može da se realizuje i bez *upsampling*-a i *downsampling*-a, postoje dve očigledne prednosti koje nosi ovakav pristup. Prva prednost ima veze sa brzinom izračunavanja i brojem potrebnih operacija. Naime, naivnom implementacijom, 3×3 konvolucija sa C filtara za ulaznu sliku dimenzija $C \times H \times W$ zahteva $9HWC^2$ operacija sabiranja i množenja što je ista cena kao da smo razmatrali 3×3 konvoluciju sa DC filtara za dimenzije ulazne slike $DC \times H/D \times W/D$. Zaključno, ukoliko se koristi *downsampling* moguće je kostiti veću mrežu za istu cenu izračunavanja.

Još jedna nova struktura koju uvode transformacione mreže su rezidualni blokovi (eng. *residual blocks*). Naime, standardne *feedforward* mreže sa porastom dubine imaju veliki problem da nauče i najprostija preslikavanja poput identiteta, usled problema sa nestajućim gradijentom (eng. *vanishing gradient*). Rezidualni blokovi veoma elegantno rešavaju ovaj problem. Jedan takav blok nalazi se na slici 12:



Slika 12. Rezidualni blok, preuzeto sa sajta https://paperswithcode.com/

Posmatrajući sliku, jasno je da uz pomoć ovakvih blokova nije potrebno naučiti željeno preslikavanje $H(x)$ već rezidual $F(x) = H(x) - x$, što mreži olakšava posao, naročito u slučajevima kada preslikavanje $H(x)$ liči na identitet.

Sa druge strane, mreža greške ima sličnu strukturu kao i kod pristupa sa optimizacijom, odnosno, uglavnom se koriste varijante VGG konvolucione neuralne mreže (na slici 11 pojavljuje se VGG-16, a ranije je pomenuto da se koristi i VGG-19).

2.3.3 Treniranje mreže

Još jedna novina koja se uvodi u odnosu na [3] i [6] je dodavanje regularizacionog člana u funkciju koja se optimizuje, te gledano sa slike 11 ona ima oblik:

$$\hat{y} = argmin_y \lambda_c l_{feat}^{\phi,j}(y, y_c) + \lambda_s l_{style}^{\phi,J}(y, y_s) + \lambda_{TV} l_{TV}(y)$$

gde je l_{TV} regularizator totalne varijanse (eng. *total variation regularizer*), a λ_{TV} njemu pripisana težina. Dodavanje ovog faktora poboljšava glatkoću slike dobijene NST algoritmom.

Tokom treninga, slika ciljnog stila y_s i struktura mreže gubitka ostaju nepromenjene. Za svaku sliku iz trening skupa ulazna slika x i slika sadržaja y_c fiksiraju na istu vrednost, vrednost slike iz trening skupa. Nakon propagacije kroz mreže i računanja odziva, na osnovu funkcije gubitka koju optimizujemo i odgovarajućih gradijenata, vrši se ažuiriranje piksela izlazne slike sve dok njen sadržaj i stil ne počnu da liče na referentne slike.

Korišćen je *Microsoft*-ov COCO skup podataka. Svaka od 80000 ulaznih slika je na početku preskalirana na dimenzije 256×256. Uzeta je veličina *batch*-a 4 i broj iteracija 40000, što približno rezultuje sa dve epohe treniranja. Korišćen je ADAM optimizator sa *learning rate*-om 1×10^{-3}. Regularizacioni faktor ima snagu između 1×10^{-6} i 1×10^{-4}. Nisu korišćeni *weight decay* ni *dropout*, s obzirom da tokom dve epohe ne dolazi do *overfit*-a. Funkcija gubitka sadržaja računata je na osnovu slojeva *relu2_2* a funkcija gubitka stila na osnovu slojeva *relu1_2, relu2_2, relu3_3,* i *relu4_3* (prema oznakama sa slike 11).

Izloženi postupak ima jako dobro vreme izvršavanja i vreme testiranja, s obzirom da kada se na ulazu mreže pojavi nova slika potreban je samo jedan *forward pass* kroz mrežu za generisanje nove slike sa zadatim stilom. Ovo je veliki napredak u odnosu na prethodne radove i ovakav algoritam je svakako moguće iskoristiti u *real-time* aplikacijama. Međutim, i ovde postoji jedno veliko ograničenje. Naime, jedna transformaciona mreže se na ovaj način specijalizuje isključivo za jedan stil, i ukoliko je potrebno izvršiti konverziju ulazne slike u nekoliko različitih stilova, potrebno je za svaki od njih istrenirati odvojenu transformacionu mrežu.

2.4 Realizacija prenosa stila primenom uslovljavanja

2.4.1 Problem

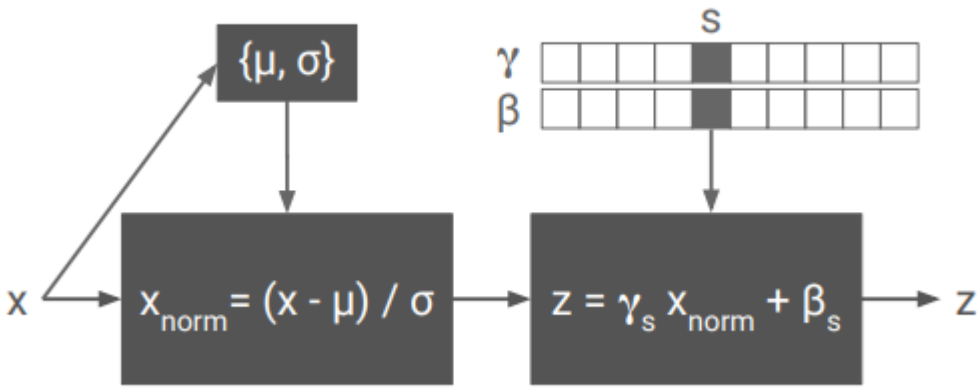
Kao što je već pomenuto, prethodna verzija NST algoritma ima ograničenje u smislu fleksibilnosti, s obzirom da je za svaki stil potrebno istrenirati novu neuralnu mrežu. Osim toga, nailazi se na još jedan veliki problem koji se tiče memorijske efikasnosti, jer bi aplikacija koja realizuje prenos stila za N različitih stilova morala da skladišti težine svake od N transformacionih mreža, što je veoma neefikasno. Takođe, treniranje zasebnih neuralnih mreža za svaki stil u potpunosti ignoriše činjenicu da veliki broj stilova može deliti iste vizualne elemente. Štaviše, stepen do kojeg bi model umetničkog stila mogao generalizuje različite stilove direktno bi izmerio mogućnost izgradnje sistema koji sažimaju karakteristike višeg nivoa i statistiku fotografija i slika.[8]

2.4.2 Normalizacija uslovne instance

Pokazuje se da veoma jednostavna modifikacija algoritma iz prethodnog poglavlja omogućava jednoj uslovnoj transformacionoj mreži da nauči da ulaznu sliku transformiše u N različitih stilova. Preostaje da se odredi kako ovo uslovljavanje treba da bude realizovano. U radu [4] prikazan je veoma interesantan rezultat koji se tiče normalizacije u NST mrežama. Kako bi se modelirao stil dovoljno je specijalizovati parametre skaliranja i pomeranja nakon normalizacije tako da odgovaraju tom specifičnom stilu. Ovaj postupak naziva se normalizacija uslovne instance (eng. *conditional instance normalization*). Cilj postupka je aktivaciju nekog sloja označenog sa x transformisati u normalizovane aktivacije z prilagođene određenom stilu. Kao što je predloženo u radu [9] parametri γ i β se proširuju u matrice dimenzija $N \times C$, gde je N broj stilova, a C broj mapa obeležja na izlazu. Uslovljavanje se vrši na sledeći način:

$$z = \gamma_s \left(\frac{x - \mu}{\sigma} \right) + \beta_s$$

gde su μ i σ srednja vrednost i standardna devijacija aktivacija x, a γ_s i β_s su određeni birajući red iz matrica γ i β koji odgovara traženom stilu. Na ovaj način omogućavamo da se jedna slika transformiše u N različitih stilova jednim jedinim prolaskom kroz *feedforward* mrežu, nasuprot prethodnim metodama, gde je za ovako nešto bilo potrebno N prolazaka. Na slici 13 prikazana je šema koja odgovara ovakvom uslovljavanju:



Slika 13. Prikaz blok šeme uslovljavanja, preuzeto iz rada [4]

Koristeći ovu metodu postižu se veoma velike uštede u broju parametara potrebnih za transformisanje ulazne slike u N različitih stilova. Ovo je glavna posledica činjenice da je najveći broj parametara ovako projektovane neuralne mreže zajednički za svih N stilova, dok je veoma mali broj parametara specifičan za pojedinačne stilove. Konkretno, od 1.6 miliona parametara mreže približno 3 hiljade parametara je specijalizovano za različite stilove (oko 0.2%). Zaključno, još jedna prednost koju nosi ovaj algoritam je veoma laka integracija novog stila. Kako bi se novi stil integrisao u mrežu potrebno je izmeniti veoma mali broj parametara mreže, koji su specijalizovani za novi stil, dok najveći broj parametara ostaje nepromenjen.

2.4.3 Arhitektura i treniranje modela

Korišćena je arhitektura transformacione mreže sa slike 14:

HYPERPARAMETERS

Operation	Kernel size	Stride	Feature maps	Padding	Nonlinearity
Network – $256 \times 256 \times 3$ input					
Convolution	9	1	32	SAME	ReLU
Convolution	3	2	64	SAME	ReLU
Convolution	3	2	128	SAME	ReLU
Residual block			128		
Residual block			128		
Residual block			128		
Residual block			128		
Residual block			128		
Upsampling			64		
Upsampling			32		
Convolution	9	1	3	SAME	Sigmoid
Residual block – C feature maps					
Convolution	3	1	C	SAME	ReLU
Convolution	3	1	C	SAME	Linear
	Add the input and the output				
Upsampling – C feature maps					
	Nearest-neighbor interpolation, factor 2				
Convolution	3	1	C	SAME	ReLU
Padding mode	REFLECT				
Normalization	Conditional instance normalization after every convolution				
Optimizer	Adam (Kingma & Ba, 2014) ($\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$)				
Parameter updates	40,000				
Batch size	16				
Weight initialization	Isotropic gaussian ($\mu = 0$, $\sigma = 0.01$)				

Slika 14. Struktura i parametri transformacione mreže, preuzeto iz rada [4]

Ovakvu strukturu mreže moguće je i modifikovati tako da sa koristi tehnika *mirror padding*-a i transponovane konvolucije (tehnika *upsampling*-a), čime bi se izbegla potreba za korišćenjem regularizacionog faktora pomenutog u prethodnim poglavljima.

Korišćen je ImageNet skup podataka kao baza za slike sadržaja. Mreža za transfer N stilova je trenirana metodom *stohastic gradient descent* koristeći ADAM optimizator. Dobijene slike su dimenzija 512×512. Kao prvi test mreža je trenirana za 10 impresionističkih slika

Claude Monet-a koje imaju relativno slične statističke parametre. Kasnije se pokazalo da je ovakva mreža sposobna da nauči još veći broj stilova bez obzira na njihovu raznovrsnost u teksturi i paleti boja. Ovi rezultati, zajedno sa rezultatima metoda navedenih ranije, nalaze se u narednom poglavlju.

3 REZULTATI

3.1 Upotreba različitih slojeva pri formiranju funkcije gubitka

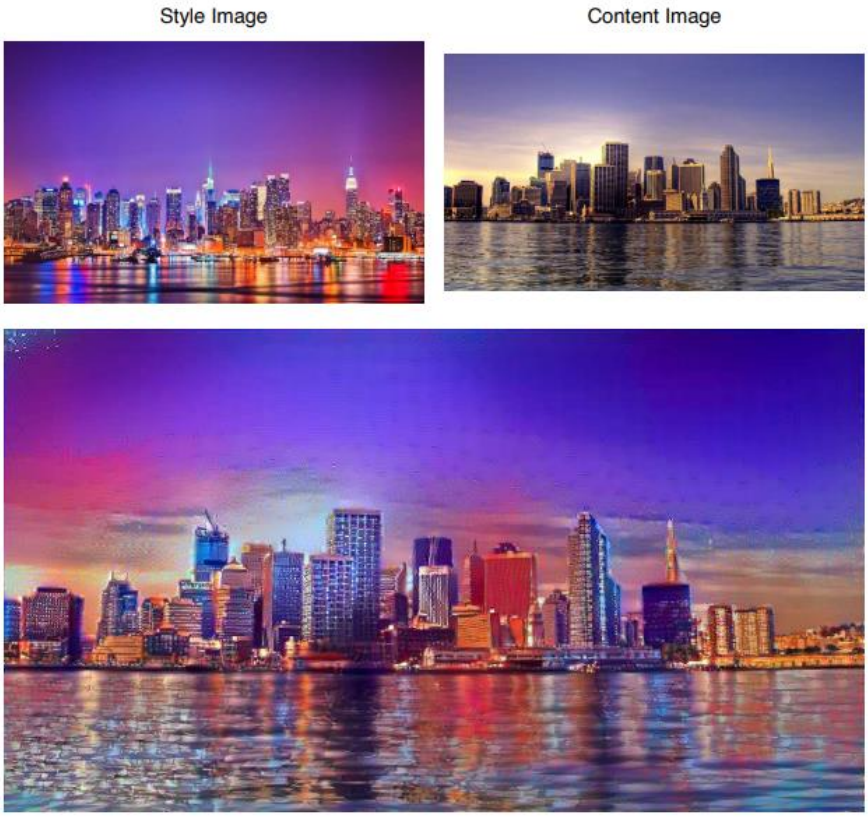
U više navrata pomenuto je da rezultati NST algoritma direktno zavise od izbora slojeva koji se koriste za formiranje funkcije gubitka. Ova pojava ilustrovana je slikom 15:

Slika 15. Poređenje različitih slojeva za formiranje funkcije gubitka, preuzeto iz rada [6]

Razmatrana slika stila je slika po imenu *Jesuiten III* čiji je autor *Lioynel Feininger*. Sa slike iznad nije teško zaključiti da izbor ranijih slojeva rezultuje finalnom slikom koja ima više detalja i jasnije definisane ivice. Sa druge strane, uzimajući kasnije slojeve finalna slika zadržava suštinu, odnosno, svi bitni objekti sa početne slike se i dalje mogu uočiti iako se veliki broj detalja praktično gubi.

3.2 Fotorealistični transfer stila

Jedan od limitirajućih faktora optimizacionog algoritma za *Neural Style Transfer* je rezolucija izlaznih slika. Složenost problema linerano raste sa brojem piksela izlazne slike, a samim tim i vreme generisanja novih slika. Takođe, sintetisane slike mogu sadržati u sebi određeni šum. Ovaj problem nije toliko značajan kada se radi o umetničkom transferu stila, međutim, ukoliko su i slika stila i slika sadržaja fotografije, izlazna slika može izgubiti na fotorealističnosti. Ipak, moguće je iskoristiti neku od tehnika za odšumljavanje kao korak postprocesiranja kako bismo dobili finalnu sliku sa boljim osobinama. Na slici 16 prikazan je primer fotorealističnog transfera stila:

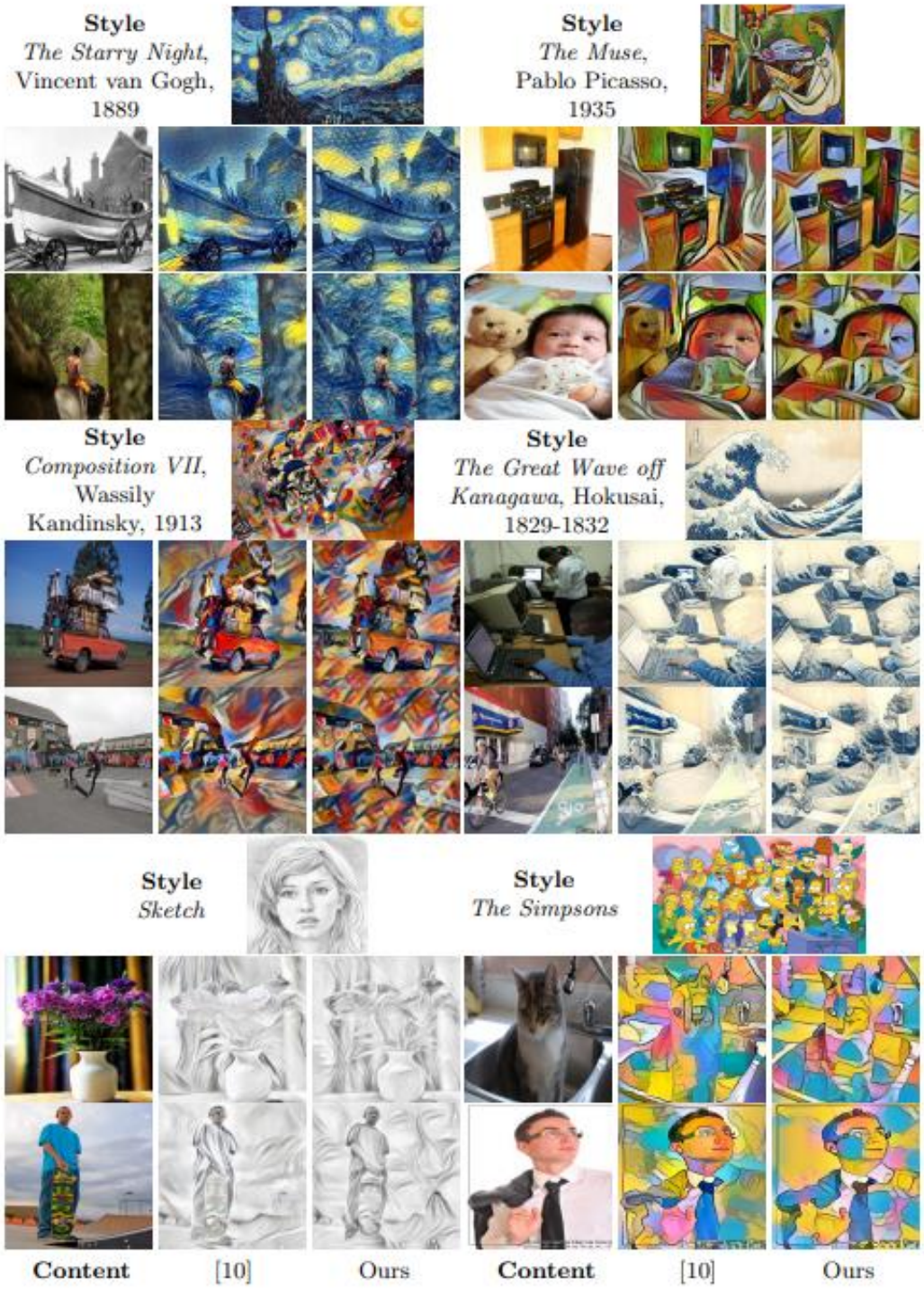


Slika 16. Fotorealistični style transfer, preuzeto iz rada [6]

Bitno je napomenuti da je veoma teško primeniti tranfer stila ukoliko karakteristični objekti sa slike stila nisu prisutni na slici sadržaja. Stoga, transfer stila se smatra uspešnim ukoliko su svi značajni objekti slike sadržaja očuvani na izlaznoj slici. Na slici iznad dobijen je poprilično dobar rezultat, s obzirom da razmatrane slike imaju gotovo isti sadržaj.

3.3 Primena transformacione mreže

Kao što je već diskutovano u poglavlju 2.2 struktura mreže za prenos stila koja koristi transformacionu neuralnu mrežu daje podjednako dobre rezultate kao i ranije korišćene metode uz značajno bolje vreme generisanja novih slika. Na slici 17 uporedo su prikazani rezultati algoritama optimizacije i transformacije za različite stilove:



Slika 17. Uporedni prikaz algoritama optimizacije i transformacije, preuzeto iz rada [7]

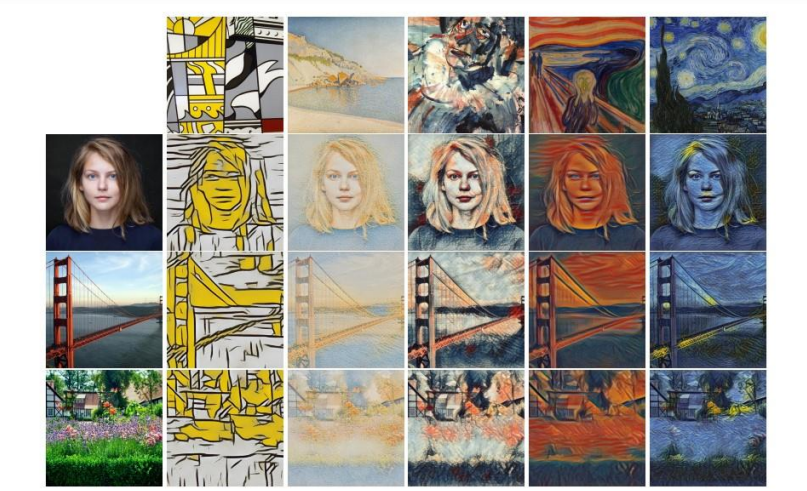
Kao glavni zaključak može se reći da obe metode daju kvalitativno iste rezultate pri čemu metoda koja koristi transformacione mreže daje rezultate neuporedivo brže, i za razliku od metode optimizacije, može se iskoristiti za potrebe *real-time* aplikacija. Na slici 18 su prikazana vremena izvršavanja za različite dimenzije ulaznih slika:

	Gatys *et al* [10]				Speedup		
Image Size	100	300	500	Ours	100	300	500
256 × 256	3.17	9.52s	15.86s	**0.015s**	212x	636x	**1060x**
512 × 512	10.97	32.91s	54.85s	**0.05s**	205x	615x	**1026x**
1024 × 1024	42.89	128.66s	214.44s	**0.21s**	208x	625x	**1042x**

Slika 18. Poređenje vremena izvršavanja za različite dimenzije izlazne slike, preuzeto iz rada [7]

3.4 Primena mreže sa uslovljavanjem

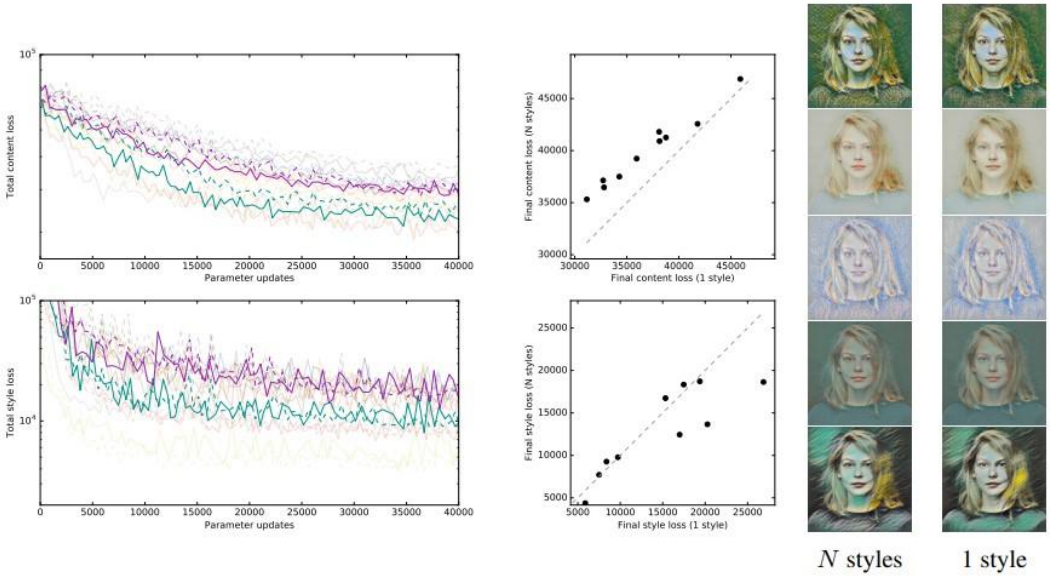
U poglavlju 2.3 uvedena je tehnika uslovljavanja sa ciljem da se omogući jednoj mreži da nauči N stilova. Pokazuje se da je pomenuta mreža u stanju da nauči čak 32 stila u isto vreme. Na slikama 18 i 19 prikazani su neki od rezultata transfera stila dobijeni korišćenjem mreže trenirane za 10 i za 32 stila:



Slika 18. Primena neuralne mreže trenirane nad 32 stila, preuzeto iz rada [4]

Slika 19. Primena neuralne mreže trenirane nad 10 stilova, preuzeto iz rada [4]

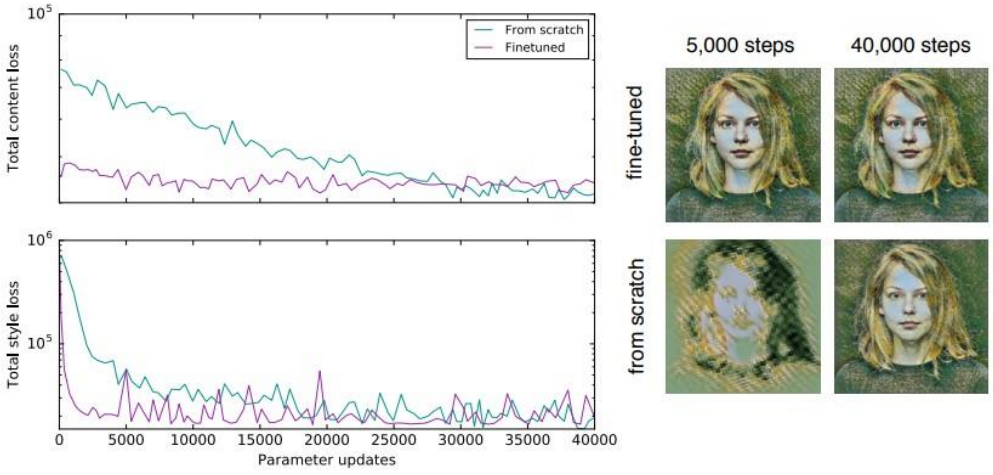
Na slici 20 prikazana je primena istog stila nad ulaznom slikom korišćenjem *feedforward* mreže trenirane za jedan stil i mreže sa uslovljavanjem, trenirane za N stilova:



Slika 20. Poređenje standardje feedforward mreže i mreže sa uslovljavanjem, preuzeto iz rada [4]

Na osnovu krivih zavisnosti funkcija gubitka od broja ažuriranja parametara mreže zaključuje se da mreža sa N stilova ima malo veću konačnu grešku sadržaja, ali veoma sličnu konačnu grešku stila. Ipak, ova razlika je gotovo neprimetna na izlaznim slikama, s obzirom da su one kvalitativno veoma slične.

Ranije je pomenuto da model sa uslovljavanjem omogućava veoma brzu i laku integraciju novog stila optimizacijom veoma malog broja parametara specifičnih za novi stil. Naime, 5000 iteracija treninga unapred istreniranog modela za uslovaljvanje sa N stilova daje rezultate sličnog kvaliteta kao i 40000 iteracija treninga potpuno nove neuralne mreže, što je ilustrovano slikom 21:



Slika 21. Uvođenje novog stila, preuzeto iz rada [4]

Mreže sa N stilova imaju mogućnost proizvoljnog kombinovanja stilova. Imajući u vidu pomenuti izraz za uslovljavanje:

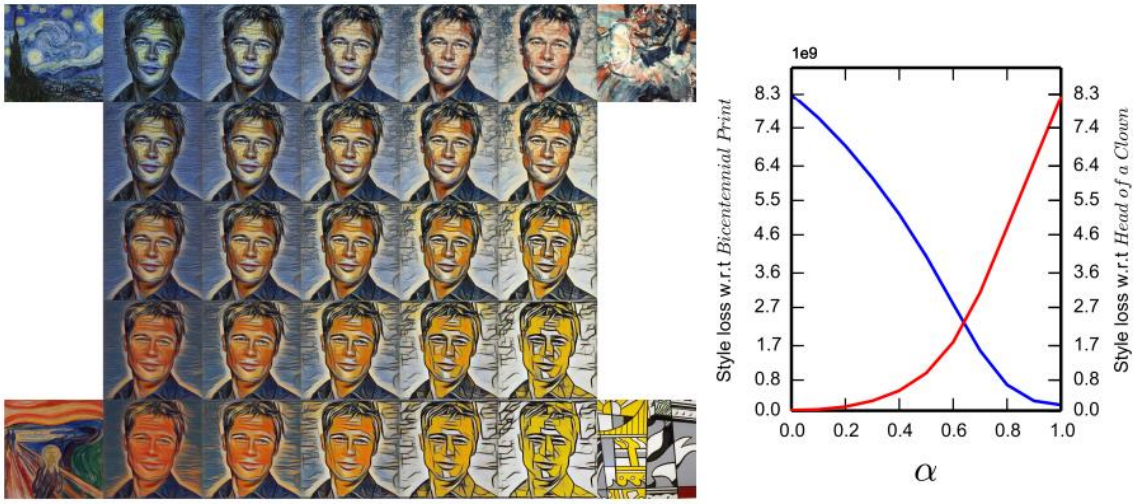
$$z = \gamma_s \left(\frac{x - \mu}{\sigma}\right) + \beta_s$$

i pod pretpostavkom da stil označen brojem 1 ima parametre (γ_1, β_1), a stil označen brojem dva parametre (γ_2, β_2), moguće je dobiti novi stil sa parametrima:

$$\gamma = \alpha \times \gamma_1 + (1 - \alpha) \times \gamma_2$$

$$\beta = \alpha \times \beta_1 + (1 - \alpha) \times \beta_2$$

Jasno je da su karakteristike novog stila direktno određene vrednošću parametra α. Za vrednosti parametra bliske 1 novi stil će biti gotovo identičan stilu 1, a za vrednosti bliske 0, identičan stilu 2. Na slici je ilustrovan je dati postupak mešanja stilova koristeći 4 različita stila:



Slika 22. Mešanje stilova, preuzeto iz rada [4]

Na slikama 23-27 prikazano je još nekoliko primera transfera stila. Slike stila su neke od slika francuskog slikara *Claude Monet*-a iz 19. veka. Slike stila nalaze se u gornjem levom uglu.

Slika 23. Grainstacks at Giverny; the Evening Sun (1888/1889), preuzeto iz rada [4]

Slika 24. Plum Trees in Blossom (1879), preuzeto iz rada [4]

Slika 25. Three Fishing Boats (1886), preuzeto iz rada [4]

Slika 26. Water Lilies (ca. 1914-1917), preuzeto iz rada [4]

Slika 27. Sunrise (Marine) (1873), preuzeto iz rada [4]

3.5 Treniranje feedforward mreže u Kaggle okruženju

Koristeći *Kaggle* platformu i *PyTorch* biblioteku za mašinsko učenje moguće je ponoviti trening *feedforward* transformacione mreže za prenos stila prema postupku navedenom u prethodnim poglavljima. Izabran je COCO-2017 skup podataka i nad njim je pokrenuta jedna epoha treninga sa sledećim setom parametara:

$$batch\_size = 10$$

$$learning\_rate = 10^{-3}$$

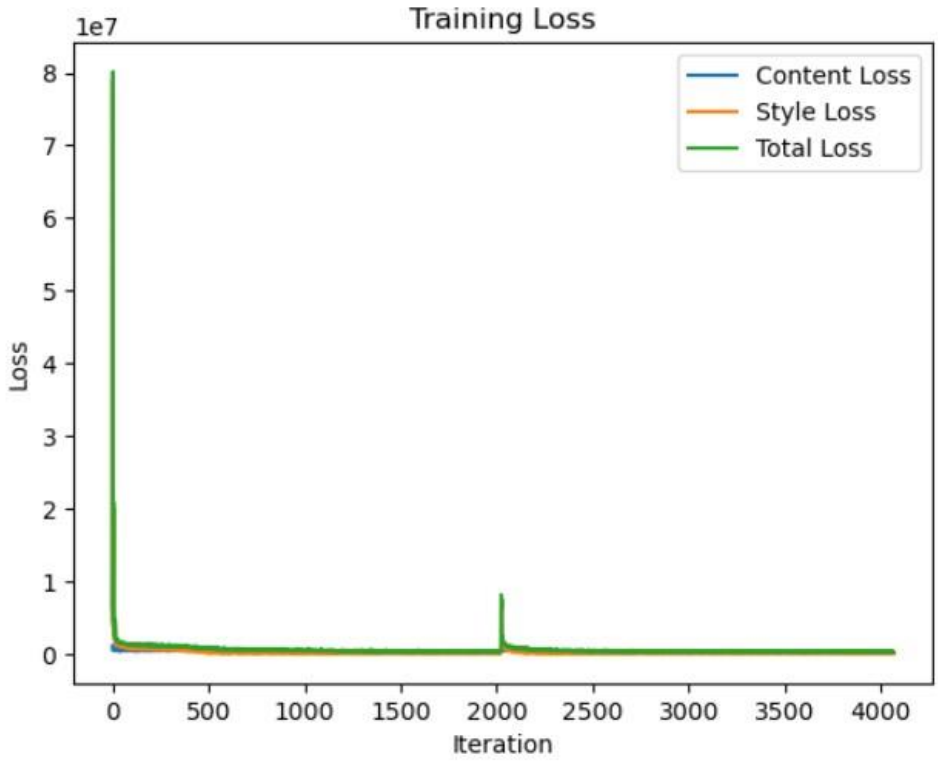
$$lambda\_style = 10^{9}$$

$$lambda\_content = 10^{4}$$

Kao i u većini slučajeva, za formiranje funkcije gubitka iskorišćena je VGG-16 mreža unapred istrenirana za klasifikaciju. Za sliku stila izabrana je čuvena *Van Gogh*-ova slika *Starry Night*. Dobijeni rezultati su prikazani na slici 28:

Slika 28. Primena istrenirane feedforward mreže

Na slici 29 prikazana je i kriva gubitka:



Slika 29. Kriva gubitka

Kod čijim pokretanjem su dobijeni ovi rezultati nalazi se na linku *https://www.kaggle.com/code/dimkela/naural-style-transfer/edit*.Važno je napomenuti da se dobijeni rezultati mogu značajno unaprediti finim podešavanjem parametara, pre svega podešavanjem broja epoha. U ovom slučaju je sprovedena samo jedna epoha treninga zbog ograničenih resursa, međutim, ukoliko se broj epoha poveća, može se poboljšati i kvalitet dobijenih rezultata.

4 ZAKLJUČAK

Na kraju, kao zaključak se može navesti činjenica da je informacije o sadržaju i stilu slike moguće izvući i tretirati potpuno nezavisno, barem za definicije sadržaja i stila navedene u poglavlju 2.2.1. U radu su detaljno objašnjene tri različite realizacije *Neural Style Transfer* algoritma: optimizacija ulazne slike, transformacija ulazne slike i transformacija ulazne slike sa uslovaljavanjem. S obzirom na to da kod ovog problema nemamo labele, već se nove slike dobijaju optimizacijom određenih kriterijuma, možemo reći da su rezultati veoma dobri, jer se subjektivnom procenom može zaključiti da zadovoljavaju početne zahteve. Pritom, dobijeni rezultati u velikoj meri zavise od sitnih modifikacija arhitekture neuralnih mreža, ali i procesa treniranja. Osim toga, uvođenjem sitnih izmena, arhitekture za prenos stila pomenute u radu mogu se iskoristiti za rešavanje brojnih drugih problema poput uvećanja rezolucije slike, kao što je navedeno u jednom od citiranih radova.

Primena NST algoritama nije ograničena samo na slike, jer se oni veoma lako mogu generalizovati na video snimke, koji zapravo predstavljaju sekvencu frejmova (slika). Velike kompanije kao što su *Instagram, Snapchat* i *Facebook* koriste transfer stila za kreiranje filtera koje transformišu slike i video zapise, što podstiče zadovoljstvo i angažovanje korisnika. Takođe, specijalizovane aplikacije poput *DeepArt* i *Prisma* omogućavaju korisnicima da svoje fotografije pretvore u slike koje liče na umetnička dela.

Generalno, zaključci izneti u ovom radu nude veliki broj mogućnosti i ideja za razvoj aplikacija baziranih na NST algoritmima. Dalja istraživanja u ovoj oblasti zajedno sa napretkom u izradi GPU tehnologije i procesora u budućnosti će omogućiti implementaciju složenijih i realističnijih stilova, što će korisnicima širom sveta pružiti neverovatne alate za kreativno izražavanje.

5 LITERATURA

[1] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. 2001. Image analogies. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH '01). Association for Computing Machinery, New York, NY, USA, 327–340. https://doi.org/10.1145/383259.383295

[2] Alexei A. Efros and William T. Freeman. 2001. Image quilting for texture synthesis and transfer. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH '01). Association for Computing Machinery, New York, NY, USA, 341–346. https://doi.org/10.1145/383259.383296

[3] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, „A Neural Algorithm of Artistic Style“, arXiv:1508.06576v2 [cs.CV] 2 Sep 2015

[4] V. Dumoulin, J. Shlens, M. Kudlur, „A Learned Representation for Artistic Style“, arXiv:1610.07629v5 [cs.CV] 9 Feb 2017

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60, 6 (June 2017), 84–90.

[6] L. A. Gatys, A. S. Ecker and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 2414-2423, doi: 10.1109/CVPR.2016.265.

[7] J. Johnson, A. Alahi, L. Fei-Fei, „Perceptual Losses for Real-Time Style Transfer and Super-Resolution“, arXiv:1603.08155v1 [cs.CV] 27 Mar 2016

[8] Simoncelli EP, Olshausen BA. Natural image statistics and neural representation. Annu Rev Neurosci. 2001;24:1193-216. doi: 10.1146/annurev.neuro.24.1.1193. PMID: 11520932.

[9] D. Ulyanov, A.Vedaldi, V. Lempitski, „Instance Normalization: The Missing Ingredient for Fast Stylization“, arXiv:1607.08022v3 [cs.CV] 6 Nov 2017