

Colorizarea pozelor alb-negru

Eric Petru Stavarache, Lucian Bicsi

Contents

0	Introducere	2
1	Munca Anterioara	2
2	Preliminarii	2
2.1	Notiuni generale	2
	Rețele generative adversariale	2
2.2	Date de antrenare	3
3	Colorizare folosind GAN-uri	3
3.1	CycleGAN	4
3.2	StarGAN	4
4	Rezultate, concluzii, posibile imbunatatiri	5
4.1	CycleGAN	5
4.2	StarGAN	6
5	Bibliografie	7

Abstract

Aceasta lucrare trateaza colorizarea pozelor folosind rețele adversariale generative, in particular arhitecturile CycleGAN si StarGAN.

Scopul lucrarii este de a testa capacitatea de invatare a rețelilor pe problema colorizarii in particular si de a o compara cu abordarile traditionale.

0 Introducere

Colorizarea pozelor alb-negru este o sarcina dificila pentru metodele traditionale de invatare.

Una dintre cele mai mari contributii la aceasta dificultate o aduce faptul ca retele traditionale sunt specializate pe minimizarea unei pierderi de tipul $\log - loss$ sau $L2$. Consideram spatiul 3-dimensional al culorilor pixelilor RGB: $\{0, 1, \dots, 255\}^3$. Punctele care minimizeaza suma distantelor $L2$ tind sa fie distribuite in jurul mediei distributiei intensitatii, astfel rezultand poze preponderent in tonuri de gri, lipsite de culori.

1 Munca Anterioara

Abordarea clasica de colorizare a pozelor este bazata pe retele convolutionale, iar rezultatele din 2016 ale lui Richard Zhang et al sunt impresionante [1][2]. Pentru a rezolva problema distributiei culorilor, autorii propun modificarea functiei de pierdere pentru a penaliza culorile care nu apar frecvent in poze. i.e., desi culoarea gri minimizeaza $L2 - loss$, nu este intalnita foarte des in pozele adevarate, deci nu va mai fi culoare dominanta in pozele colorizate.

2 Preliminarii

2.1 Notiuni generale

Retele generative adversariale

Definiție 2.1. Retelele generative adversariale (eng. Generative Adversarial Network) au fost inventate de catre Ian Goodfellow in 2014. Acestea sunt cea mai noua metoda de a rezolva problema esantionarii dintr-un spatiu al datelor unde abordarile traditionale au un bias catre medie. [5].

Aceste retele sunt compuse din doua parti: reseaua generatoare si reseaua discriminatoare. Reteaua generatoare G incearca sa genereze output din distributia \mathcal{O} , bazandu-se fie pe input din distributia \mathcal{I} . Vom nota distributia lui G cu \mathcal{G} . Reteaua discriminatoare D consta intr-un **clasificator (binar)** care primeste ca input o poza $i \in \mathcal{I}$ si incearca sa distinga daca i a provenit din \mathcal{O} , distributia adevarata a datelor, sau din \mathcal{G} , distributia pe care o genereaza G .

Indicele de performanta al retelei G este data de cat de bine reuseasca sa "pacaleasca" reseaua D , in timp ce performanta retelei D este cat de bine reuseste sa distinga intre datele

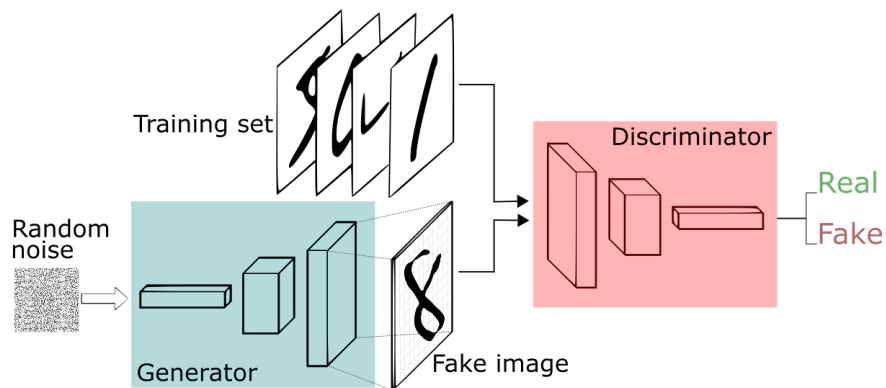


Figure 1: Retele Adversariale Generative

adevarate si cele artificiale. Retelele vor fi antrenate in paralel, in decursul a mai multor epoci.

Motivatia centrala din spatele acestui tip de model isi are originile in **teoria jocurilor**. Retelele G si D pot fi vazuti ca doi adversari intr-un joc de tip min-max: G incearca sa maximizeze scorul, in timp ce D incearca sa minimizeze scorul. Aici scorul este definit ca fiind, spre exemplu, numarul de poze clasificate gresit de catre D . De aici provine numele de RETELE ADVERSARIALE.

2.2 Date de antrenare

Pentru antrenarea retelei am combinat dataset-urile puse la dispozitie de autorii CycleGAN [6]. Aceste dataset-uri au fost menite pentru a transforma dintr-un domeniu specific in altul (de exemplu de a converti din cai in zebre). Pentru a spori diversitatea, am creat datasetul prin reuniunea tuturor pozelor puse la dispozitie, deoarece domeniul de poze color include toate celelate poze. Pentru fiecare poza din reuniune, am generat varianta alb-negru.

3 Colorizare folosind GAN-uri

Aceasta lucrare incearca sa trateze problema colorizarii folosind Retele Adversariale Generative - GAN. Abordarea naiva ar fi formata din cele doua componente clasice: retea generatoare si cea discriminatoare. Reteaua generatoare G primeste o imagine alb-negru si genereaza una color. Reteaua discriminatoare D primeste o imagine color, si incearca sa distingă daca este o imagine naturala, sau daca a fost generata de catre D .

Principala problema cu aceasta abordare este ca retea generatoare descopera ca optim este sa genereze mereu aceeasi poza colorata, fara a tine cont de input. Pentru a rezolva asta, vom utiliza generatori bidirectionali, idee prezentata in cadrul a doua abordari arhitecturale:

prima dintre ele se numeste *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*, sau CycleGAN [3][4], iar a doua *Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation* sau StarGAN [7][8].

3.1 CycleGAN

CycleGAN este un model care rezolva problema ca output-ul generatorului nu depinde de input. Ideea centrala este sa avem un generator $G1$ care ne trece din domeniul \mathcal{I} in domeniul \mathcal{G} si un generator $G2$ care ne trece din \mathcal{G} in \mathcal{I} . Functiile lor de pierdere vor fi corelate de faptul ca vrem ca transformarea inversa sa fie idempotentă. i.e., daca din alb-negru trecem o poza in color si apoi inapoi in alb-negru, vrem ca cele doua poze alb-negru sa fie cat de similare se poate.

Evaluarea performantei celor doi generatori va fi facuta de catre doi discriminatori specializati $D1$ si $D2$. Astfel, $D1$ va evalua poze color adevarate vs. poze color false generate de catre $G1$, iar $D2$ va evalua poze alb-negru adevarate vs. poze alb-negru false generate de catre $G2$.

CycleGAN a obtinut rezultate remarcabile in cadrul altor task-uri asemanatoare computational cu cel al colorizarii, precum transformarea cailor in zebre, transformarea peisajelor de vara in iarna, zi - noapte, etc.

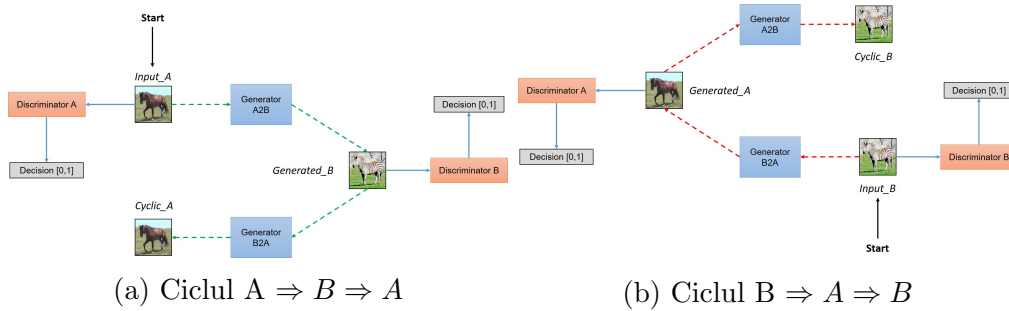


Figure 2: Modelul de functionare CycleGAN

3.2 StarGAN

StarGAN este o retea generativa, construita in original pentru transferul atributelor faciale si pentru a incerca sa reproduca expresii/gesturi faciale corespunzatoare sentimentelor primare (fericire, tristete, dezgust, surprindere, etc.). Strategia este diferita de cea din spatele CycleGAN, in sensul ca prezinta un generator cu functie multipla, in locul celor doi generatori $G1$ si $G2$ mutual inversi ai retelei CycleGAN. Acelasi lucru se intampla si in cadrul discriminatorilor, fiind imbinati intr-unul singur, care nu mai este binar. Arhitectura

generatorului este de tip **encoder-decoder**, in timp ce discriminatorul este un clasificator binar peste o retea convolutionala.

Diferenta principala intre functionalitatile StarGAN comparativ cu CycleGAN este ca StarGAN trateaza problema mai multor clase posibile de domenii, ceea ce nu ne este util in cazul colorizarii propriu-zise, inasa se poate folosi daca dorim diverse tipuri de colorizari (sepia, noapte/zi, tonuri calde/reci, etc.). Cu toate acestea, pentru acest lucru este nevoie de un dataset mult mai mare si mai expresiv, de care nu dispunem momentan.

Un mare avantaj in cadrul implementarii pytorch a StarGAN [8] este faptul ca retea este puternic customizabila, putand seta hiperparametri precum redimensionarea imaginilor input, setarea numarului de straturi intermediare in cadrul generatorului / discriminatorului, (astfel avand posibilitatea de a testa codul local pe o arhitectura minimala inainte de antrenarea propriu-zisa), setarea intervalului de iteratii la care este salvat modelul, capacitatea de a porni de la un model preantrenat, etc.

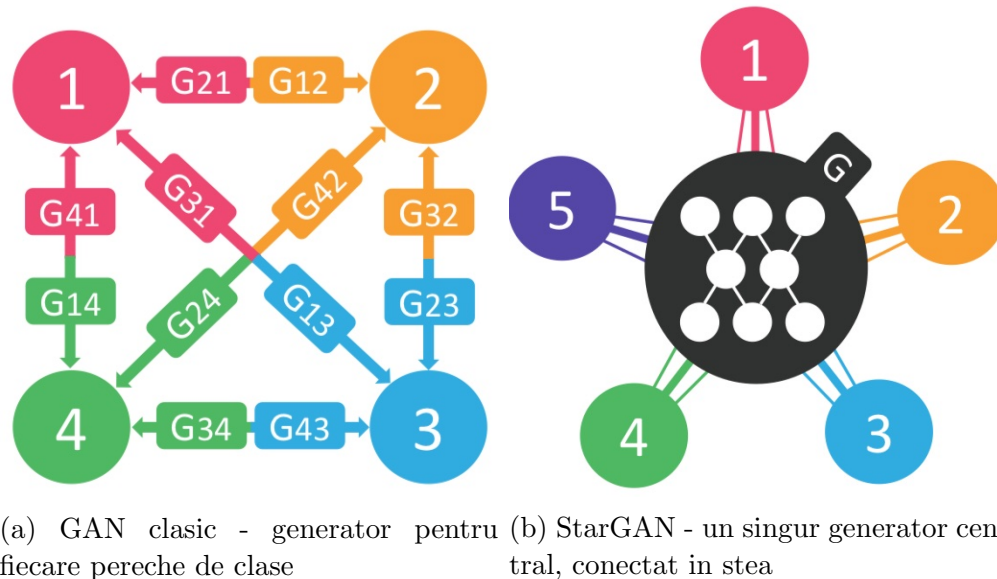


Figure 3: Modelul de functionare StarGAN

4 Rezultate, concluzii, posibile imbunatatiri

4.1 CycleGAN

Pozele colorizate de catre retea CycleGAN sunt satisfacatoare. Comparativ cu metodele de colorizare state-of-the-art, outputul retelei pare incetosat. Acest fapt se genereaza cantitati mici de date de antrenare folosite, iar o posibila imbunatatire ar fi utilizarea setului de antrenare ImageNET.

O alta imbunatatire generala posibila consta, bineinteles, in antrenarea mai multor iteratii si eventual in antrenarea unor retele mai adanci pe o cantitate mai mare de date si cu un numar mai mare de iteratii.

Un alt mod prin care performantele pot fi imbunatatite sunt prin calibrarea mai buna a hiperparametrilor pe problema specifica a colorizarii.

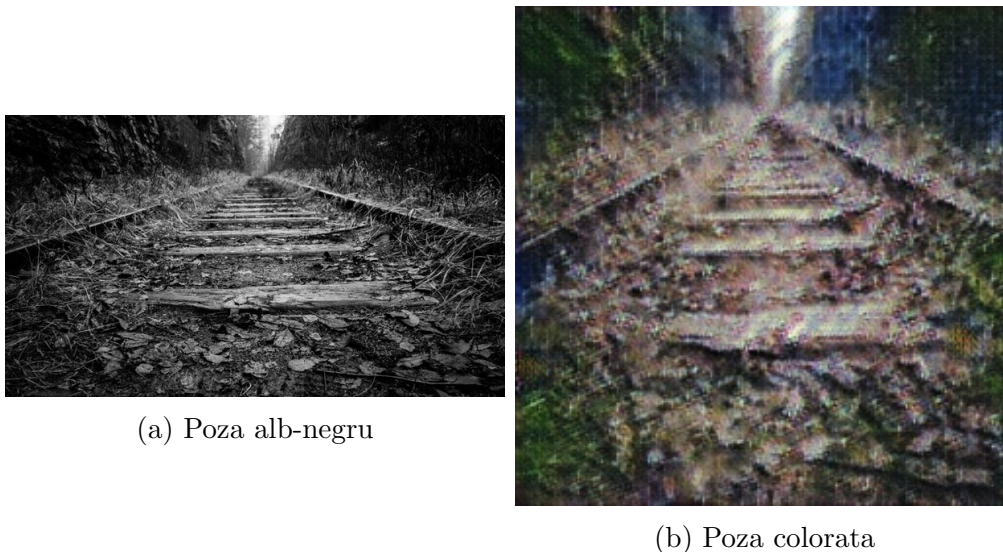


Figure 4: Colorizarea unei poze cu CycleGAN

4.2 StarGAN

Spre deosebire de rezultatele obtinute de reseaua CycleGAN, reseaua StarGAN, spre surprinderea noastra, **nu a obtinut rezultate** dupa un numar mare de iteratii. In configuratia default, reseaua nu colorizeaza poze alb-negru si nu decolorizeaza poze color, ci ajunge sa reproduca imaginile din input in ambele cazuri.

Ipoteza noastra este ca problema transferului atributelor faciale, in cadrul careia o transformare [*happy* \Rightarrow *sad*] si una [*sad* \Rightarrow *happy*] impart o cantitate foarte mare din informatiile imaginii input (i.e., restul imaginii ramane neschimbat), fapt ce determina ca un generator "shared" sa reuseasca sa codifice informatiile esentiale cu un numar mic de parametri, in timp ce problema colorizarii formalizat in acelasi cadru prezinta diferente uriase la nivel de structura a pozei. In cadrul problemei colorizarii, discriminatorul StarGAN trebuie sa invete dubla functionalitate de colorizare si decolorizare aproximativ independent in cadrul acelorasi parametri.

5 Bibliografie

- [1] Richard Zang, Phillip Isola, Alexei A. Efros
Colorful Image Colorization, <https://arxiv.org/pdf/1603.08511.pdf>, ECCV, 2016.
- [2] Richard Zang *Colorful Image Colorization*, <http://richzhang.github.io/colorization/>
- [3] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros, *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*, <https://arxiv.org/abs/1703.10593>
- [4] Jun-Yan Zhu, *CycleGAN Project Page*, <https://junyanz.github.io/CycleGAN/>
- [5] Ian J. Goodfellow et al
Generative Adversarial Networks, <https://arxiv.org/abs/1406.2661>
- [6] Taesung Park, *CycleGAN datasets*,
https://people.eecs.berkeley.edu/~taesung_park/CycleGAN/datasets/
- [7] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, Jaegul Choo, *StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation*,
<https://arxiv.org/abs/1711.09020>,
- [8] Yunjey Choi *StarGAN: PyTorch implementation*,
<https://github.com/yunjey/StarGAN>