

# Generování obrázků maleb z fotografií

Luboš Zápotočný

ČVUT - FIT

zapotlub@fit.cvut.cz

1. prosince 2022

## 1 Úvod

GAN se skládá ze dvou neuronových sítí: model generátoru a model diskriminátoru.

Generátor je (hluboká) neuronová síť, která vytváří obrázky. Pro daný dataset se snažím generovat obrázky ve stylu malíře Moneta. Tento generátor je trénován pomocí diskriminátoru.

Tyto dva modely pracují proti sobě, přičemž generátor se snaží oklamat diskriminátor a diskriminátor se snaží přesně klasifikovat skutečné vs. generované obrázky.

Cílem je vybudovat GAN, který generuje 7000 obrázků ve stylu malíře Moneta z reálných fotografií.

Transformaci z reálné fotografie na obrázek, který se bude co nejvíce podobat malířskému stylu použiji jeden z modelů pro tzv. style transfer, konkrétně CycleGAN.

Transformace obrázku na obrázek je problém, kde je cílem naučit se mapování mezi vstupním obrázkem a výstupním obrázkem pomocí trénovací sady zarovnaných párů obrázků. Zarovnané obrázky jsou takové, u kterých máme k dispozici zdrojový obrázek a také příslušný obrázek po transformaci. Bohužel takovýto dataset není vždy možné vytvořit. Například při transformaci zimní krajiny na letní bychom museli vyfotit mnoho míst v zimním období a na úplně stejném místě za 6 měsíců vytvořit ideálně úplně stejnou fotku. U jiných příkladů již takové mapování ani není možné vytvořit. Například u obrázků, které namalovali malíři před stovkami let se ani neví, kde dané namalované místo může být.

Tvorba datasetu je tedy buď časově nebo peněžně velmi nákladná a v některých případech i nemožná. CycleGAN umožňuje vytrénovat model hlubokého učení, který se dokáže tuto transformaci naučit i bez zarovnaných (spárovaných) dat. [6], [8], [7], [1], [5], [4], [2], [3]

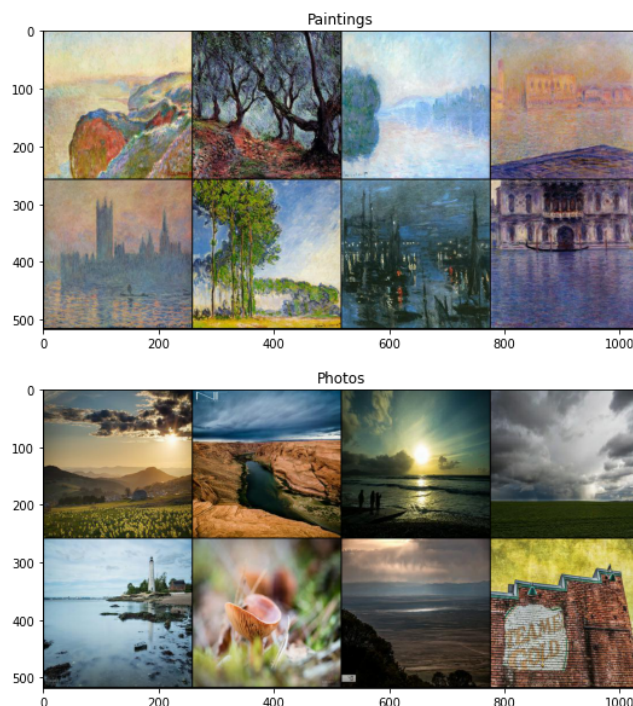
## 2 Vstupní data

Vstupní data jsem převzal ze soutěžního datasetu na webové stránce Kaggle.com

Dataset obsahuje 300 obrázků (256x256 pixelů) některých částí obrazů od malíře Moneta. A obsa-

hují více než 7000 fotografií různého druhu u kterých se očekává převedení stylu malby Moneta na tyto fotografie.

Ukázka datasetu je vyobrazena na obrázku 1.



Obrázek 1: Ukázka vstupních dat

## 3 Metody

Jako první bych chtěl zmínit, že bylo nutné přibližně skoro 10 hodin řešit výkon nahrávání obrázků z Google disku, který byl připojen do běhového prostředí Google Colab. Takto vytvořené síťové připojení disku mělo velmi negativní dopad na rychlost načítání dávek (batches) při nahrávání dat pro trénování modelů.

Tento problém byl vyřešen efektivním archivačním algoritmem, který mi dovolil vstupní data (obrázky) uložit na vzdálené úložiště (github repozitář, archiv má < 100MB) a při inicializaci trénovacího Notebooku data stáhnout, extrahovat a používat jako lokální složku v běhovém prostředí. Efektivita čtení obrázků (z disku) se násobně zrychlila.

Výzkum CycleGAN zmiňuje použití ResNET ar-

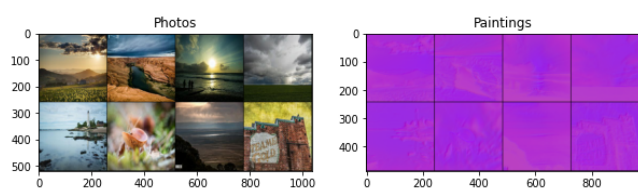
chitektury, kterou budu chtít porovnat s UNet architekturou.

## 4 Výsledky

Výsledný style transfer není dokončený. Chybí implementovat trénovací část a díky experimentálnímu vyhodnocení ladit parametry modelů a případně struktury jednotlivých komponent.

Prozatím jsem se soustředil na detailní rešerši o obsáhlých tématech. Zároveň jsem se snažil pochytit best-practices s prací s PyTorch.

Obrázek 2 vyobrazuje první spuštění generátoru na vstupních datech. Zatím není aplikované učení pomocí diskriminátoru, ale i tak je vidět, že konvoluční vrstva zafungovala.



Obrázek 2: Testování generátoru (bez trénování)

## 5 Závěr

Základní struktura trénovacích komponent byla vytvořena a byl vyřešen problém pomalého načítání z Google disku na běhovém prostředí Google Colab.

Práce prozatím není dokončená a chybí jí trénovací fáze modelu. Každopádně základní komponenty generátorů a diskriminátorů jsou již v první verzi připravené a budou v následujících iteracích vylepšovány tak, aby obsahovali co nejvíce poznatků z výzkumu CycleGAN a dosahovaly uspokojivých výsledků.

Rád bych tuto práci využil jako reprezentativní práci v nadcházejících projektech a chtěl bych vytrénovaný model vyzkoušet na real-time převodu videa.

## Reference

- [1] Gan (generative adversarial network): Simple implementation with pytorch, Jun 2022.
- [2] Jason Brownlee. A gentle introduction to batch normalization for deep neural networks, Dec 2019.
- [3] Jason Brownlee. A gentle introduction to batch normalization for deep neural networks, Dec 2019.

- [4] Jason Brownlee. How do convolutional layers work in deep learning neural networks?, Apr 2020.
- [5] Jason Brownlee. How to develop a cyclegan for image-to-image translation with keras, Sep 2020.
- [6] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.
- [7] Diego Gomez Mosquera. Gans from scratch 1: A deep introduction. with code in pytorch and tensorflow, May 2020.
- [8] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2017.