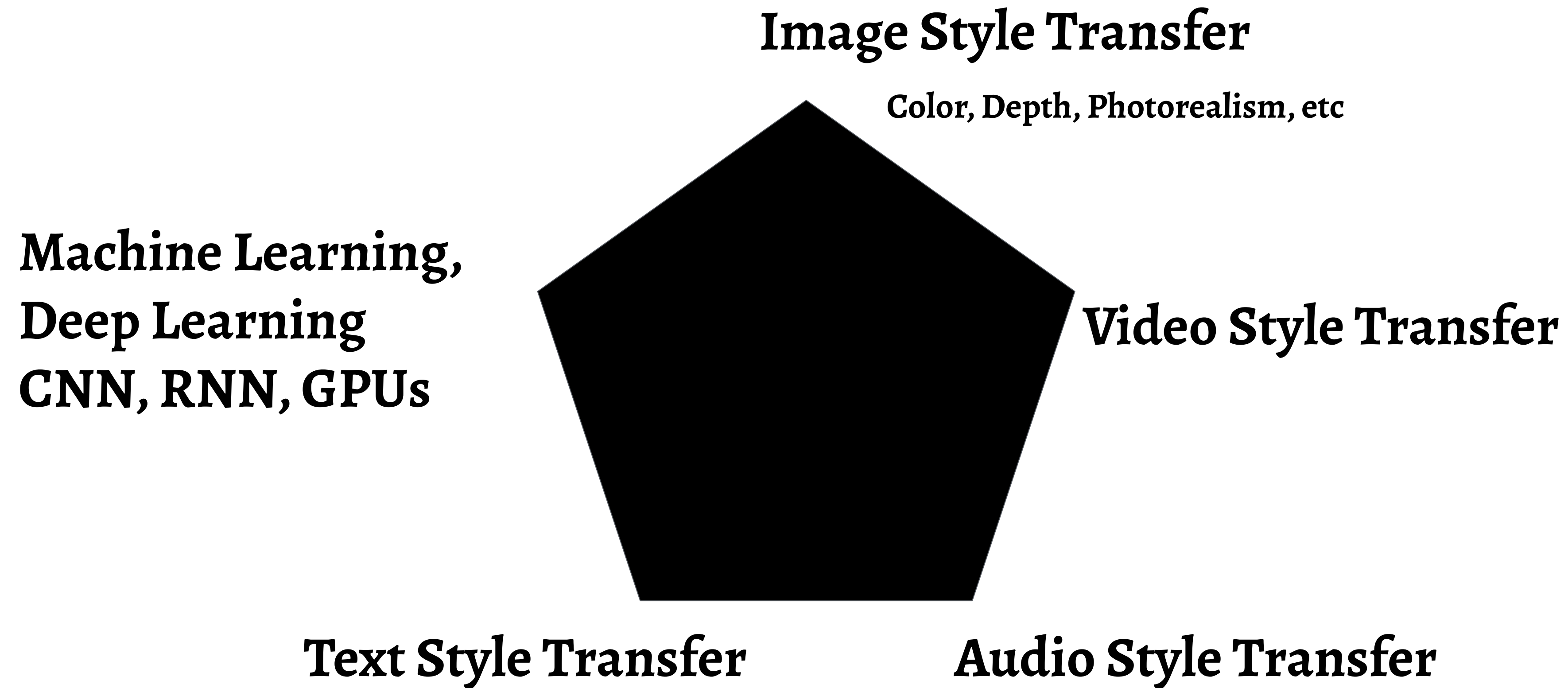




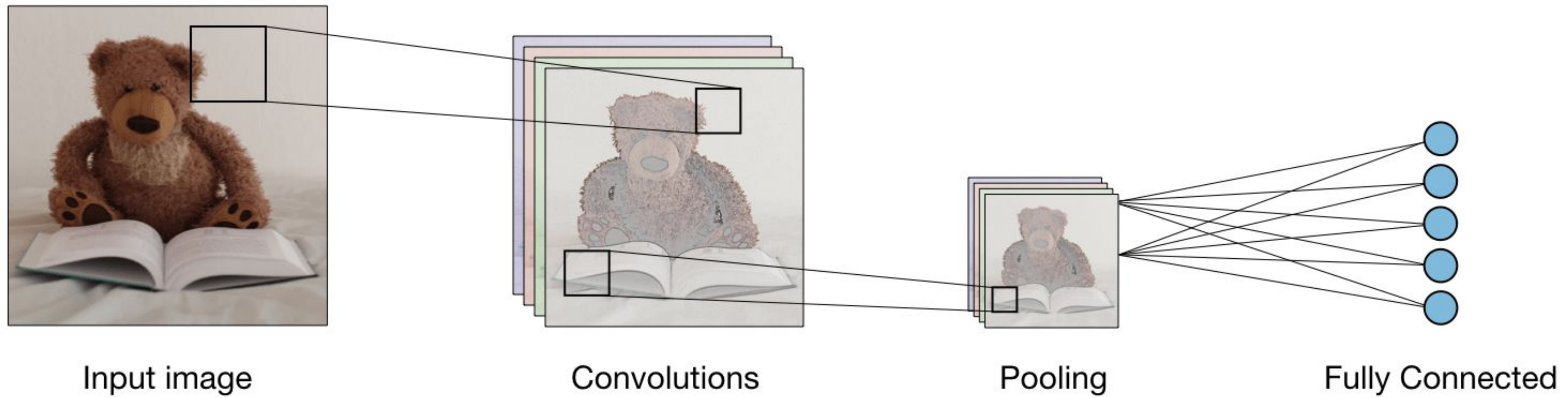
Высшая Школа Экономики
Московский Институт Электроники и Математики

**Разработка алгоритма
преобразования стиля
изображений с применением
алгоритмов машинного обучения
и параллельных вычислений**

Павел Цветов, БИВ 161



Convolutional Neural Networks



Целью данной работы является разработка алгоритма “Нейронного переноса стиля” для рендеринга изображений с высоким разрешением, с использованием алгоритмов оптимизации и ускорения вычислений в нейронной сети и параллельных распределенных вычислений.

1. Рендеринг изображений с высоким разрешением (Full HD Up to 4K)
2. Рассмотрение возможных способов
3. Параллельные и распределенные вычисления, алгоритмы, методы, использование вкупе с Cloud GPUs
4. Проведение экспериментов с Color Preservation, Depth/Perspective Preservation, Brush Size/Stroke, MultiStyle, PhotoRealistic Style
5. Эксперименты с гиперпараметрами модели (learning rate, optimizer, etc)

- Deep Learning (the early idea — 60s - 80s; real use — 2000s)
- Convolutional Neural Networks (CNN) — emerged in 80s; real use in 2000s
- Original paper: Gatys et al. (2015) — “Image Style Transfer Using Convolutional Neural Networks”
- Johnson et al. (2016) — “Perceptual Losses for Real-Time Style Transfer”
- Multiple Style Transfer and Video Style Transfer (2017)

- Вердикт: obsolete research papers, major flaws, various limitations to our cause, not reflecting on our issue

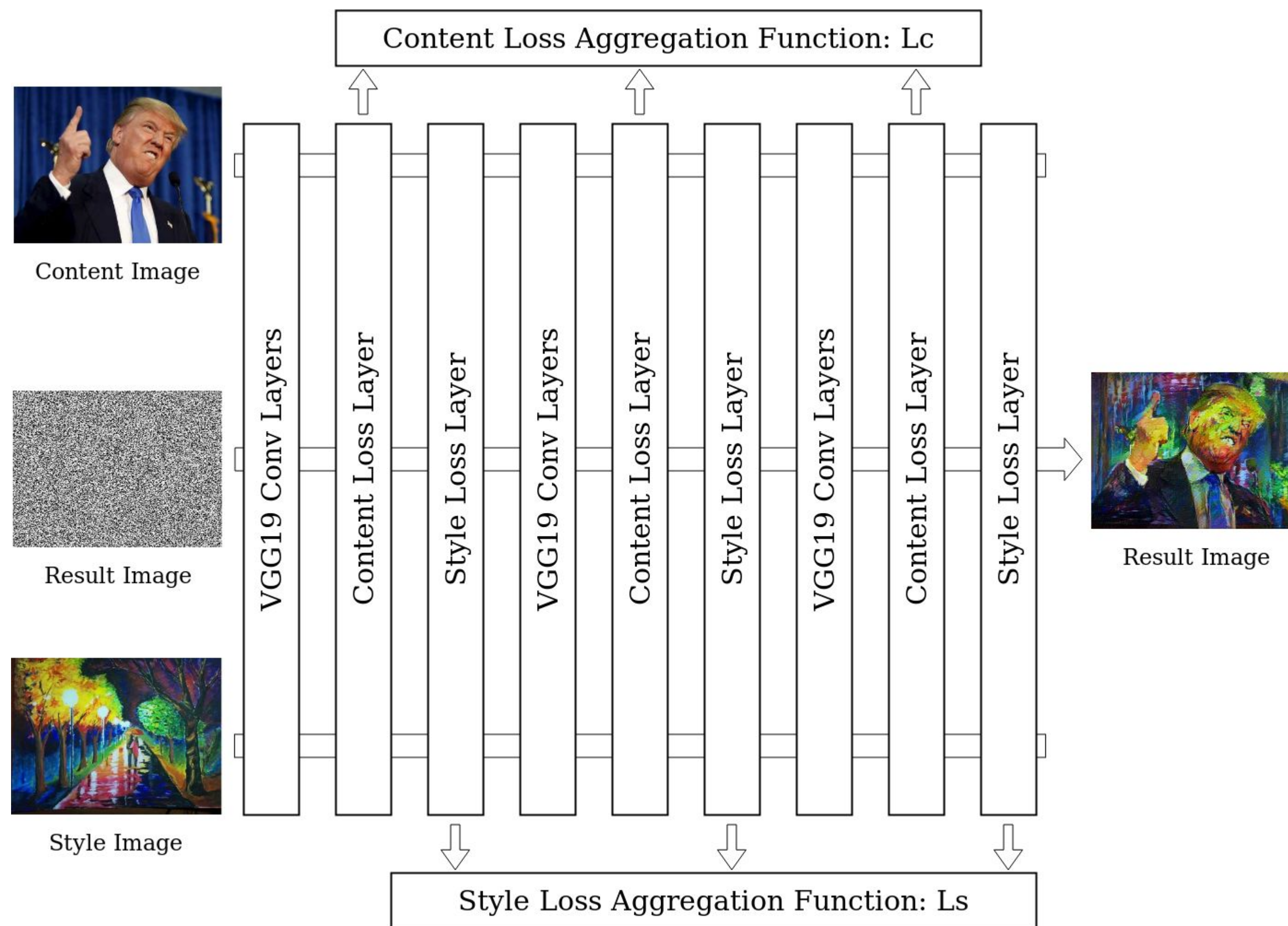
Постановка задачи



- Input: given Content image Y_C and Style image Y_S generate output image Y , so that:

$$Y = \operatorname{argmin}(y) L_{\text{total}}(Y_C, Y_S, Y)$$

- Issue: Baseline Method Image Size to VRAM (1920x1080 ~ 12 GB)
- How to generate high-resolution images (Full HD up to 4K) ?
- How to change and control perceptual factors (e.g. brush size & stroke, color, texture, etc.) ?



$$L_{\text{total}} = \alpha L_c + \beta L_s + \gamma L_i$$

1. Implementation of original Gatys et al. Algorithm
2. Extending a baseline algorithm adding new features (color, depth, perspective, stroke)
3. Render of High-Resolution images:

Super Resolution Image Upscaling, Image to Tile Splitting

4. Experimental stage

Tools:

- Python 3 language, user-friendly PyTorch ML framework
- Cloud GPU Platforms, Jupyter Notebook, Google Colab, etc
- Using pre-trained CNN VGG19 network, external NNs

Фактические результаты / стадия проекта




1. Обзор литературы и изучение работы алгоритма Neural Style Transfer
2. Реализация алгоритма Гэтиса нейронного переноса стиля с использованием библиотеки PyTorch
3. Ознакомление с облачными GPU-платформами для обучения модели и настройки рабочей среды
4. Расширение реализованного алгоритма для рендеринга изображений с высоким разрешением, рассмотрение способов (склейка, Super Resolution Upscaling, и т.д.)
5. Реализация алгоритма нейронного переноса стиля с сохранением исходного цвета

<https://github.com/yaiestura/neural-style-transfer>

Colab Google



 PhotoRealisitic.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

+ Code + Text

RAM
Disk

Editing

```
[7] from PIL import Image
import matplotlib.pyplot as plt
import numpy as np
import time

import torch
import torch.nn as nn
import torch.optim as optim
import torchvision
from torchvision import transforms, models

IN_MOMENTUM = 0.1

from google.colab import drive
drive.mount('/content/gdrive')
```

Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).

```
[9] device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print(device)

cuda

[10] class ReflectionConv(nn.Module):
    """
    Reflection padding convolution
    """
    def __init__(self, in_channels, out_channels, kernel_size, stride):
        super(ReflectionConv, self).__init__()
        reflection_padding = int(np.floor(kernel_size / 2))
        self.reflection_pad = nn.ReflectionPad2d(reflection_padding)
        self.conv = nn.Conv2d(in_channels, out_channels, kernel_size, stride)

    def forward(self, x):
        out = self.reflection_pad(x)
        out = self.conv(out)
        return out
```


Нейронный перенос стиля



Нейронный перенос стиля



Нейронный перенос стиля



Нейронный перенос стиля



Недоработки

1. Реализация алгоритма нейронного переноса стиля с сохранением глубины и перспективы исходного изображения
2. Multi Style Transfer
3. Изучение факторов, влияющих на толщину и размер кисти на генерируемом изображении
4. Проведение различных экспериментов, улучшающих общее качество и эффективность алгоритма
5. Изучение алгоритмов параллельных вычислений для использования в комбинации с алгоритмом нейронного переноса стиля

Развитие

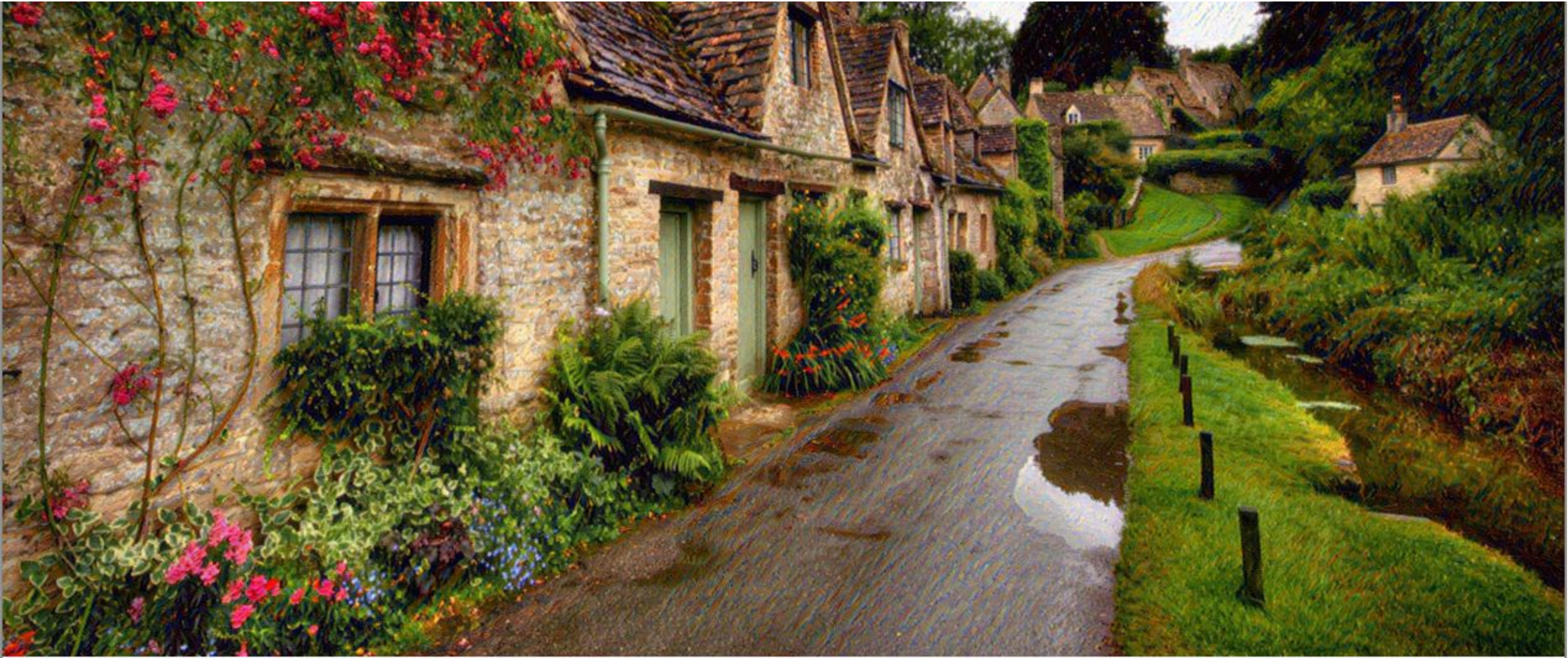


- Generative Adversarial Networks (GAN)
- Real-Time Arbitrary Style Transfer
- *Video Style Transfer*
- *Shading*
- VRAM efficiency
- *Automatic* hyperparameter tuning

Нейронный перенос стиля



Super Resolution



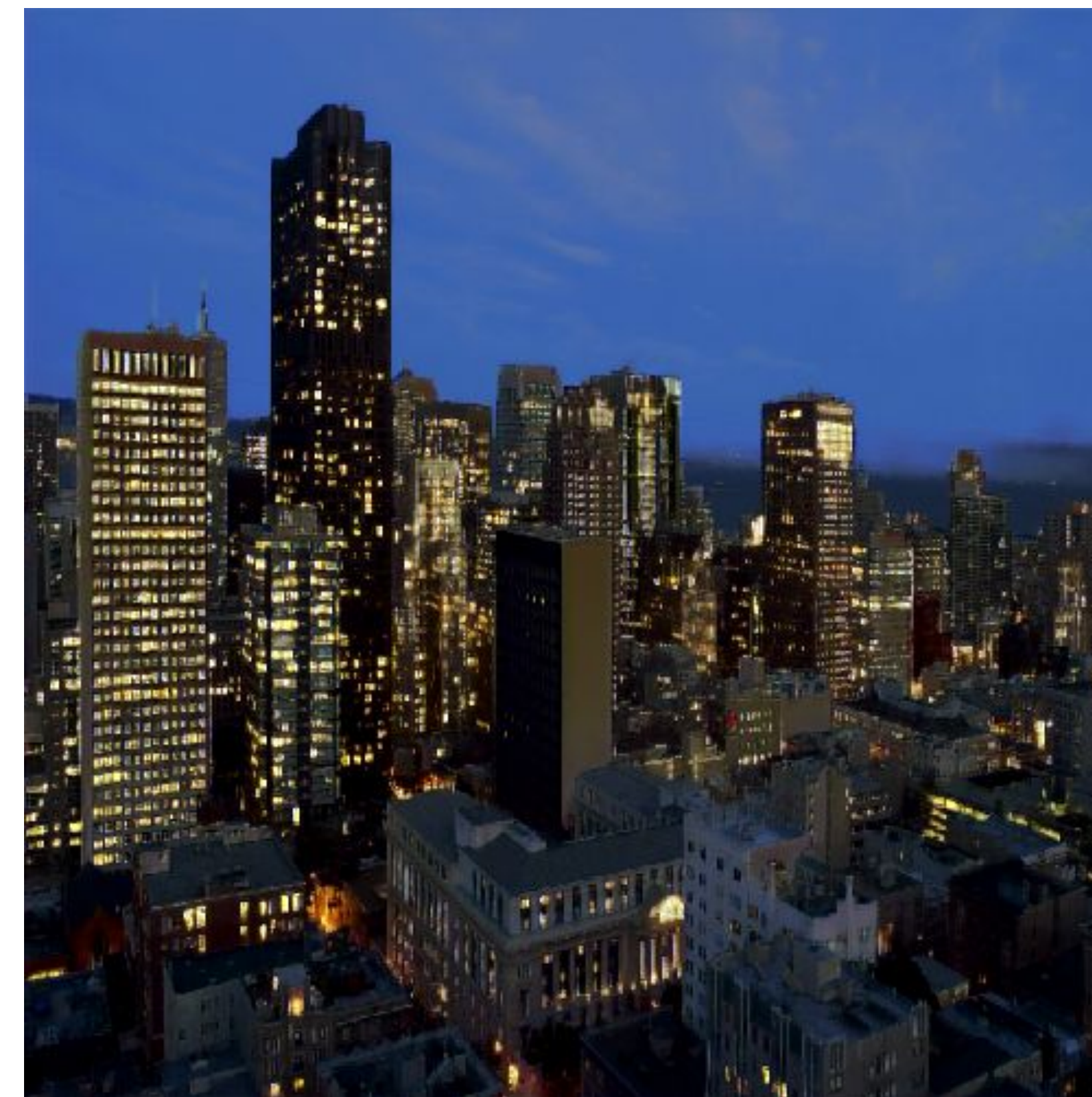
Сохранение цвета



Сопоставление гистограмм (Histogram Color Matching) и Перенос по яркости (Luminance-only transfer)



Фотореалистичный перенос стиля



Single-Image Depth Perception in the Wild, 2017



Input image

Our Depth

Zoran

Eigen

Ground Truth