Исследование применения циклических генеративно-состязательных нейронных сетей для стилизации изображений

Д.И. Ульянов[1], Д.А. Савельев[1,2]

[1]Самарский национальный исследовательский университет им. академика С.П. Королева, Московское шоссе 34А, Самара, Россия, 443086
[2]Институт систем обработки изображений РАН - филиал ФНИЦ «Кристаллография и фотоника» РАН, Молодогвардейская 151, Самара, Россия, 443001

Аннотация. В работе проведены примеры архитектур свёрточных нейронных сети, соответствующих функций активаций и организация их взаимодействия в процессе обучения. Сети взаимодействуют друг с другом согласно архитектуре генеративно-состязательных сетей. Для задачи был отфильтрован и отформатирован набор данных NEXET 2017. Были проведены исследования архитектур нейронных сетей и варьирования объёма тренировочной выборки для решения задачи стилизации изображений.

1. Введение

Последнее время индустрия обработки изображений является одной из самых быстрорастущих в мире. Киностудии нанимают сотни дизайнеров, чтобы укладываться в сроки создания CGI (Computer-GeneratedImagery) видео. Выполняемая же работа требует больших затрат времени. Данные свойства характеризуют её как потенциального кандидата на то, чтобы заменить её работников искусственными нейронными сетями (ИНС). Особенно сложным и монотонным отмечается процесс стилизации, а в частности превращение дневной картинки в ночную и наоборот. До 2014 года данная проблема не имела решения, позволяющего проводить стилизацию в приемлемом для нынешнего поколения форматов изображений разрешении (1280×720 или 1920×1080 пикселей). В 2014 году была изобретена архитектура генеративно-состязательной сети (англ. GenerativeAdversarialNetworks, GAN) [1].В 2016 году была опубликована работа, предлагающая новую архитектуру на основе GAN – циклическая генеративно-состязательная сеть (англ. Cycle-Consistent Generative Adversarial Network, cycleGAN) [2]. Для её обучения достаточно иметь изображения, разделенные на множества, которые обобщены по некоторому ряду признаков. При этом в результате обучения получается сразу два стилизатора. Благодаря использованию данного метода можно выявлять специальные характеристики одного множества изображений и определять, как они могут быть переведены в другое множество, и все это при отсутствии каких-либо парных обучающих примеров.

Эта проблема может быть более широко описана как задача стилизации – преобразование изображения из одного представления данной сцены X в другое, Y [3]. Годы исследований в области компьютерного зрения, обработки изображений, компьютерной фотографии и графики

привели к созданию мощных систем перевода в контролируемой среде, где, например, доступны пары изображений.

Однако получение парных данных обучения может быть сложным и дорогостоящим. Например, существует только пара наборов данных для таких задач, как семантическая сегментация, и они относительно малы [4]. Получение пар ввода-вывода для графических задач, таких как художественная стилизация, может быть еще более сложным, поскольку, как правило, требует художественной разработки. Предположим, что между множествами существуют некоторые базовые отношения, например, что они представляют собой два разных стиля рисовки одной и той же базовой сцены и стремимся изучить эти отношения. Хотя нам не хватает контроля в виде парных примеров, мы можем использовать контроль на уровне наборов: нам предоставляется один набор изображений в области X и другой набор в области Y. Мы можем обучить отображение X в Y так, чтобы выход был неотличим от изображений из Y противником, обученным отличить подделки от оригинала.

В данной работе были рассмотрены архитектуры GAN и cycleGAN. Была реализована искусственная нейронная сеть на основе архитектуры cycleGAN, адаптирован набор исходных данных под задачи обучения и тестирования. Так же были проведены эксперименты с целью обнаружить оптимальные параметры сети для решения задачи стилизации, в сравнении с эталонным результатом одной из реализаций cycleGAN.

2. Архитектуры и функции потерь для нейронных сетей

Согласно архитектуре GAN, в процессе обучения участвует две сети – генератор и дискриминатор. Генератор можно представить, как 3 блока: блок выделения признаков, блок преобразования признаков и восстановления данных по признакам. Дискриминатор по своей сути является бинарным классификатором и обладает соответствующей данной задаче архитектурой. В данной работе используются следующие слои: свёрточный (conv), остаточный блок (ResNet) и блок развёртки (TransConv). Свёрточный слой производит выделение признаков. Остаточный блок состоит из двух соединённых свёрточных слоёв, при этом на выходе из блока добавляются данные с входа на слой, что снижает последствия проблемы деградации сети, когда при большом числе слоёв качество результата работы сети уменьшается. Слой развёртки на основе поданных на вход признаков воспроизводит данные, обладающие данными признаками. Архитектуры генераторов и дискриминаторов приведены в таблицах 1 и 2.

Ввиду крайней вычислительной сложности, обусловленной характеристиками тестовой системы, функции ошибок принимают следующий вид:

$$D_A^{loss} = (D_A(a) - 1)^2 + D_A(G_{B\to A}(b))^2, \quad (1)$$

$$D_B^{loss} = (D_B(b) - 1)^2 + D_B(G_{A\to B}(b))^2, \quad (2)$$

где $D_A(a)$ и$D_B(b)$ – функции дискриминаторов картинок классов A и B соответственно. Введём функцию циклической ошибки для перехода из одного класса в другой:

$$C_A^{loss} = \frac{1}{3 \times L \times Z} \sum_{i=1}^{L} \sum_{j=1}^{Z} \sum_{k=1}^{3} \left| a_{ijk} - \tilde{a}_{ijk} \right|, \quad (3)$$

где a_{ijk} и$\tilde{a}_{ijk}$– пиксели из i-стоки, j-столбца, k-цветового канала изображения a, поданного на генератор $G_{A\to B}$и$\tilde{a} = G_{B\to A}(G_{A\to B}(a))$соответственно.

Общая функция циклической ошибки с учётом (3) приобретёт следующий вид:

$$C^{loss} = \frac{1}{2}(C_A^{loss} + C_B^{loss}). \quad (4)$$

Итоговые функции ошибки генераторов с учётом (4) принимают следующий вид:

$$G_{A\to B}^{loss} = (D_B(G_{A\to B}(a)) - 1)^2 + \lambda C^{loss}, \quad (5)$$

$$G_{B\to A}^{loss} = (D_A(G_{B\to A}(b)) - 1)^2 + \lambda C^{loss}. \quad (6)$$

Таблица1. Архитектура генератора.

Тип слоя	Размер входного слоя	Размер ядра	Расстояние между свёртками	Функция активации
Conv	256×256×3	3×3	2	ReLU
Conv	128×128×32	3×3	2	ReLU
Conv	64×64×64	3×3	2	ReLU
Conv	32×32×128	3×3	2	ReLU
ResNet	32×32×128	3×3	1	ReLU
ResNet	32×32×128	3×3	1	ReLU
ResNet	32×32×128	3×3	1	ReLU
ResNet	32×32×128	3×3	1	ReLU
ResNet	32×32×128	3×3	1	ReLU
TransConv	32×32×128	3×3	2	ReLU
TransConv	64×64×64	3×3	2	ReLU
TransConv	128×128×32	3×3	2	ReLU
Conv	256×256×16	3×3	1	ReLU
Out	256×256×3	-	-	-

Таблица2. Архитектура дискриминатора.

Тип слоя	Размер входного слоя	Размер ядра	Расстояние между свёртками	Функция активации
Conv	256×256×3	4×4	4	ReLU
Conv	64×64×32	4×4	4	ReLU
Conv	16×16×64	4×4	4	ReLU
Conv	4×4×128	4×4	4	Softmax
Out	1×1	-	-	-

3. Тренировочные данные для нейронной сети

Для обучения нейронной сети была выбрана часть базы данных NEXET 2017, являющаяся фотографиями с авто-регистраторов в разрешении 1280×720[11]. База данных содержит 17000 изображений.

Главная проблема NEXET 2017 в нашем случае – это отсутствие готового разделения на дневной и ночной наборы. Для разделения на дневные и ночные наборы было отобрано вручную 200 изображений (по 100 на каждый класс), и была высчитана яркость каждого из них по следующей формуле яркости из HSP модели [12]:

$$L = \sqrt{0,299 \times r^2 + 0,587 \times g^2 + 0,114 \times b^2}, \quad (7)$$

где r, g и b – дискретные значения RGB-составляющих пикселя в диапазоне [0; 255]. После все значения были нормализованы делением на максимальное значение. Результаты приведены на рисунке 1.

Значения нормированной яркости были отсортированы для того, чтобы скачок яркости был нагляден. На рисунке виден разрыв между графиками на интервале нормированной яркости [0,48; 0,52], из чего можно вынести предположение, что приблизительно в этом интервале присутствуют изображения, которые невозможно классифицировать как дневные или ночные. Ввиду того, что при столь малом интервале разделения высока вероятность ложного определения, расширим данный интервал до [0,4; 0,6]. Будем разделять на дневной и ночной наборы по следующему правилу: если можно установить границу нормированной яркости, выше которой изображения можно классифицировать как дневное, а ниже, соответственно, как ночное. После классификации всей базы данных по такому правилу было получено 4 695 ночных и 11 442 дневных изображения, из которых в тестовую выборку было отобрано по 500 случайных изображений на класс. Ввиду крайне ограниченных ресурсов для подобного рода

задач для нейронных сетей, каждое изображение было обрезано до соотношения сторон 1:1 и масштабировано до 256×256 пикселей. Примеры из тренировочной выборки приведены на рисунке 2.

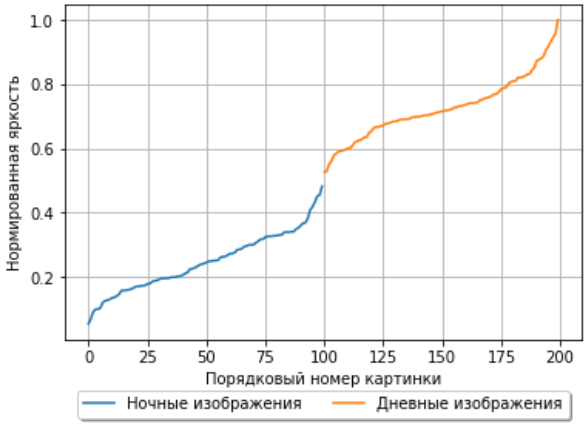

Рисунок 1. График нормированной яркости изображения.

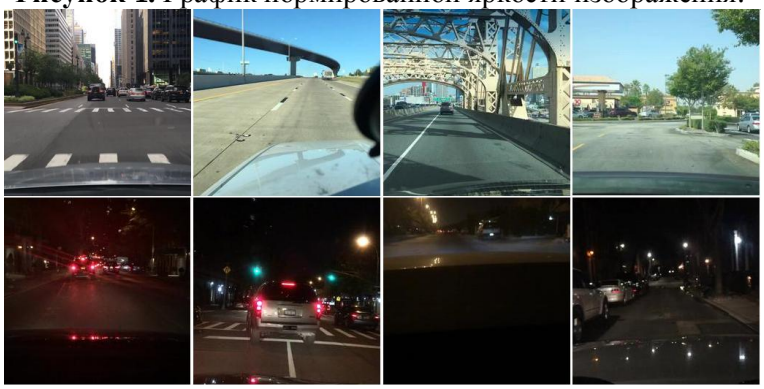

Рисунок 2. Примеры тренировочной выборки.

4. Проведение экспериментальных исследований и анализ результатов

Параметры эксперимента:

- процессор – intel core i5-2500, 3,3 ГГц;
- оперативная память – 8Gb, 1333 МГц;
- графический процессор – Nvidia GTX 1060, 6Gb GDDR5;
- операционная система – Windows 10 x64. Версия 1903;
- язык – Python 3.7.1 x64.

Была написана программа, реализующая алгоритмы обучения ИНС и стилизации изображений с помощью спроектированной модели с применением следующих библиотек:

- NumPy – библиотека для работы с массивами.
- Pillow – библиотека для работы с изображениями.
- Tensorflow-gpu 1.13.1 – библиотека для работы с ИНС через графический процессор.

Нейронная сеть обладает следующими параметрами:

- скорость обучения (learning rate) $\alpha = 2\times10^{-4}$ для всех сетей;
- размерность принимаемых изображений – 256×256×3;
- функция активации нейронов – ReLU;
- коэффициент циклической потери$\lambda = 10$.

Ввиду того, что оптимальный размер тренировочной выборки для данной задачи стилизации неизвестен, будем руководствоваться числами, применяемыми при обучении таких стилизаторов как лошадь ↔ зебра и яблоко ↔ апельсин, то есть 1000 элементов из каждого класса. Размер тестовой выборки – 100 элементов. Число эпох – 100. График ошибок дискриминаторов и генераторов приведён на рисунке 3.

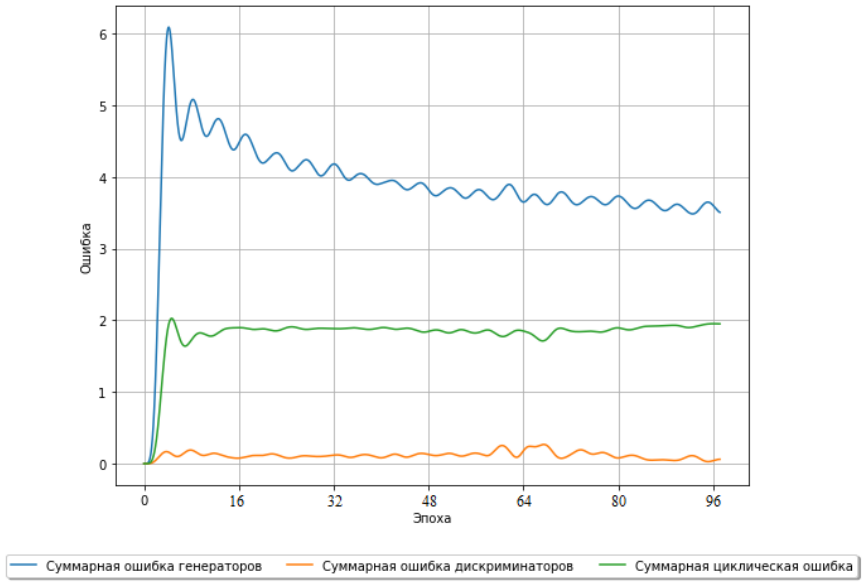

Рисунок 3. Графики ошибоксети, обученной на неполной выборке.

Результаты преобразования тестовых изображений приведены на рисунках 4 и5.

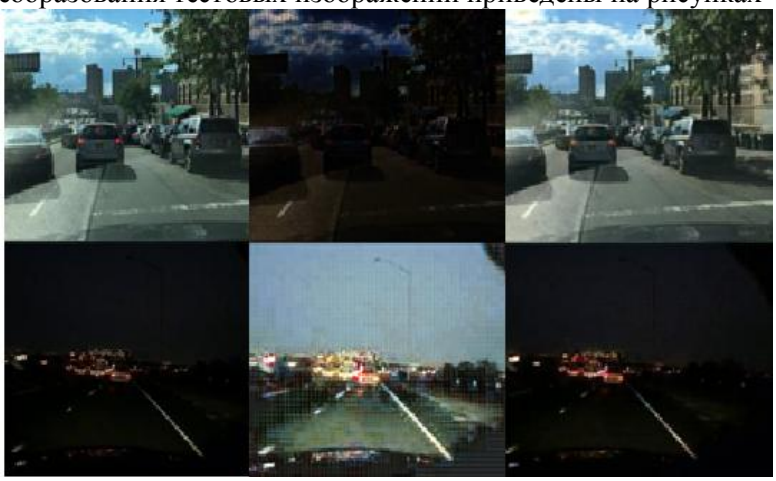

Рисунок 4. Результат стилизации изображения.

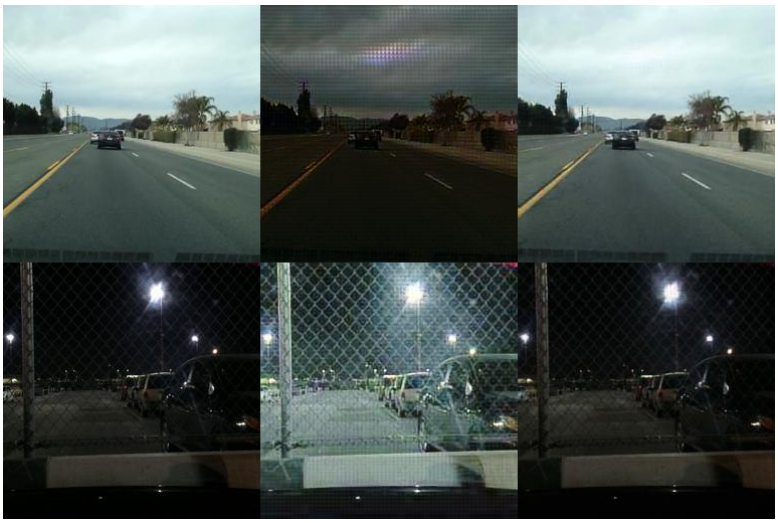

Рисунок 5. Примеры преобразований, когда основные признаки не были выделены.

Исследования показали, что преобразование из дневного в ночное не всегда работает корректно – повышается лишь яркость некоторых элементов сцены, в то время как небо не стилизуется. Увеличим число тренировочных примеров до 10000 дневных и 4500 ночных. По 500 изображений из каждого класса используются в качестве тестовых примеров. Из-за резко возросшего объёма данных уменьшим число эпох до 20. График ошибок дискриминаторов и генераторов приведён на рисунке 6.

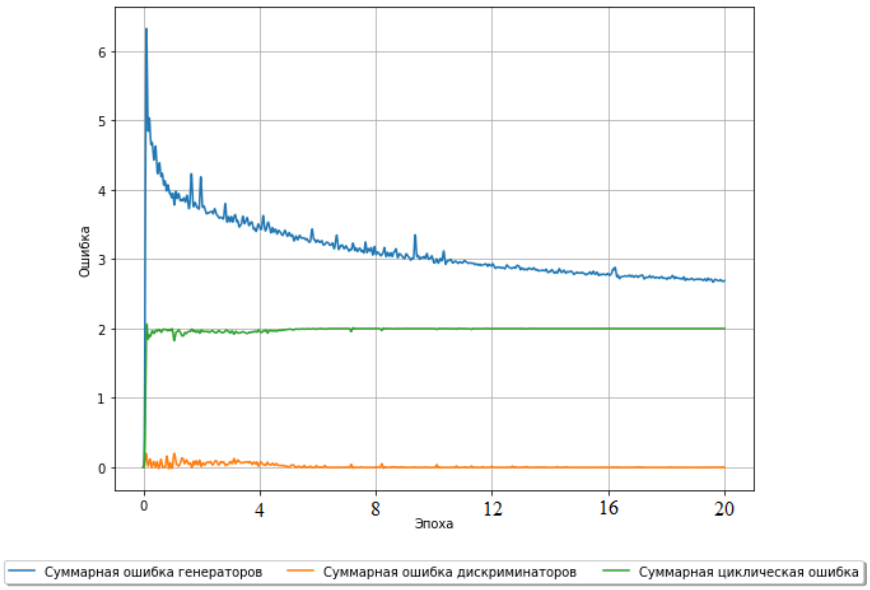

Рисунок 6. Графики ошибоксети, обученной на полной выборке.

Результат преобразования тестовых изображений приведен на рисунке 7.

Рисунок 7. Результат стилизации изображения при увеличении числа тренировочных примеров.

Изменим скорость обучения на несколько порядков с целью узнать, является ли стандартная для большинства задач скорость обучения $\alpha = 2{\times}10^{-4}$ приемлемой для данной задачи. Проверим на одном и том же тестовом изображении результат обработки нейронных сетей, обученных на неполном (1000 изображений в каждом классе) наборе данных с разными показателями скорости обучения: $2{\times}10^{-4}$, $2{\times}10^{-3}$ и $2{\times}10^{-2}$. Число эпох – 20. Результат эксперимента приведён на рисунке 8.

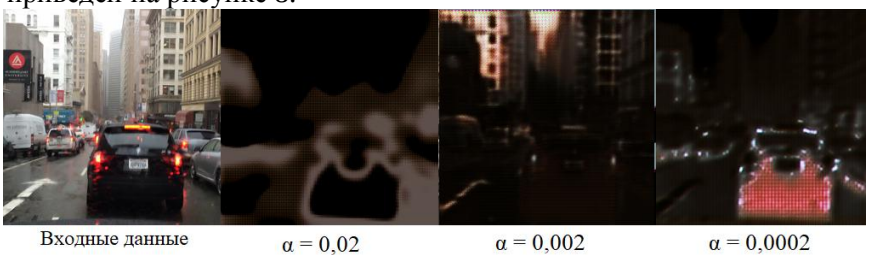

Рисунок 8. Зависимость результата от скорости обучения.

Данный результат подтверждает, что стандартная скорость обучения является приемлемой для поставленной задачи.

Сравнивая рисунки 5 и 7 можно заметить то, что полученная в результате первого эксперимента нейронная сеть либо не подставляет признак источников света ночью, либо не декодирует его в полной мере, так как на рисунке 8 заметны проявления этого признака, что говорит о необходимости большого числа эпох для данной задачи стилизации.

5. Заключение

В рамках исследования был разработан программный комплекс для демонстрации работоспособности архитектуры cycleGAN в задачах стилизации изображений. Сформированы тренировочная и тестовая выборки из набора данных NEXET 2017.

В ходе работы были решены следующие задачи: реализована архитектура cycleGAN, сформирована база данных для обучения и тестирования, обучена ИНС на полном и неполном наборе тренировочных данных.

Исследования показали, что для решения задачи стилизации изображений под дневные и ночные стилистики с помощью ИНСследует максимизировать число уникальных элементов тренировочной выборки. Это позволяет уменьшить результат суммы функций потерь на 25% при меньшем числе эпох, что говорит об улучшении качества стилизации. Показано, что для данной задачи в выбранной конфигурации ИНС оптимальной скоростью обучения является 2×10^{-4}.

6. Литература

[1] Goodfellow, I. Generative adversarial nets / I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio // Advances in neural information processing systems, 2014. – P. 2672-2680.

[2] Zhu, I. Unpaired image-to-image translation using cycle-consistent adversarial networks / J.Y. Zhu,T. Park, P. Isola, A. A. Efros // Proceedings of the IEEE international conference on computer vision, 2017.– P. 2223-2232.

[3] Isola, P. Image-to-image translation with conditional adversarial networks / P. Isola, J. Y. Zhu, T. Zhou, A.A. Efros // Proceedings of the IEEE conference on computer vision and pattern recognition, 2017. – P. 1125-1134.

[4] Xiaolong, L. Recent progress in semantic image segmentation / L. Xiaolong, Z. Deng, Y. Yang // Artificial Intelligence Review. – 2019. – Vol. 52(2). – P. 1089-1106.

[5] Рашид, Т. Создаем нейронную сеть / Т. Рашид – СПб.: Альфа-книга, 2017. – 272 с.

[6] Nitta, T. Solving the XOR problem and the detection of symmetry using a single complex-valued neuron / T. Nitta // Neural Networks. – 2003. – Vol. 16(8). – P. 1101-1105.

[7] Kriesel, D. A Brief Introduction to Neural Networks, 2005.

[8] Saha, S. A Comprehensive Guide to Convolutional Neural Networks – the ELI5 way // Medium, 2018.

[9] Naoki, S. Up-sampling with Transposed Convolution // Medium, 2017.

[10] Li, B. An improved ResNet based on the adjustable shortcut connections / B. Li, Y. He // IEEE Access. – 2018. – Vol. 6. – P. 18967-18974.

[11] NEXET Dataset, 2017 [Электронный ресурс]. – Режим доступа: https://www.kaggle.com/solesensei/nexet-original (20.01.19).

[12] HSP Color Model–Alternative to HSV (HSB) and HSL [Электронный ресурс]. – Режим доступа: http://alienryderflex.com/hsp.html (30.01.19).

The investigation of the using the cyclic generative-competitive neural networks for image stylization

D.I. Ulyanov[1], D.A. Savelyev[1,2]

[1]Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086
[2]Image Processing Systems Institute of RAS - Branch of the FSRC "Crystallography and Photonics" RAS, Molodogvardejskaya street 151, Samara, Russia, 443001

Abstract. The paper provides examples of convolutional neural network architectures, the corresponding activation functions, and the organization of their interaction in the learning process. Networks interact with each other according to the architecture of generative-adversarial networks. For the task, the NEXET 2017 data set was filtered and formatted. Studies of the architecture of neural networks and varying the volume of the training sample to solve the problem of image styling were carried out.