Обзор статьи «Deep Photo Style Transfer» (Luan et al.)

Почему именно эта статья?

В 2015 году статья «A neural algorithm of artistic style» (Gatys et al.) показала как сверточные нейронные сети могут быть использованы для генерации изображений, имеющих содержание одной картинки, а стиль другой. Особенно успешно получилось переносить стиль с картин на фотографии — как в популярном приложение Prisma. Но переносить стиль таким образом, чтобы генерируемое по фотографии изображении оставалось фотореалистичным не получалось.

Исследователи из Корнелля и компании Адоб справились с этой трудностью. Теперь правильно подобрав образец для переноса стиля, можно «изменять» время суток, освещение, погоду или художественную обработку своей фотографии. Результат статьи стал вирусным: получил ≈ 11000 звёздочек на гитхабе и ≈ 1100 голосов на HackerNews:

> This is super impressive and something that I didn't think would be possible without someone very skilled in photoshop going over the images.

Кроме того, у этой статьи нет реализации на моём любимом PyTorch, а авторская реализация — мясорубка из матлаба и луа. Правда есть, но не очень опрятная реализация на TensorFlow.

Краткий обзор нейросетевого переноса стиля (aka ELIA5)

См. диаграмму на следующей странице!

Подробнее про статьи

A neural algorithm of artistic style (Gatys et al.)

В этой работе впервые исследовался вопрос как с помощью свёрточных сетей переносить стиль с известных картин (далее S) на данную фотографию (далее I). Много информации о стиле и семантическом содержании изображения содержится в активациях предобученной CNN. В качестве статистики стиля выступают активации из первых свёрточных слоёв глубокой CNN, а в качестве статистики содержания — из последних. Такое разделение обязанностей между слоями соответствует нашей интуиции о предназначении свёрток на разных слоях и работах по визуализации фич свёрточных сетей.

Ключевая идея — итерационно улучшать изображение, грубо говоря, делать backpropagation в картинку, оптимизируя сумму стилевой и контекстуальной целевых функций.

В качестве контекстуальной функции потерь выбрана норма разницы между функциями активации оптимизируемого изображения (далее O) и I. В качестве стилевой функции потерь выбрана норма разницы матриц Грама активаций O и S. [1]

$$\mathcal{L}_{total} = \sum_{\ell} \underbrace{\mathcal{L}^{\ell}_{context}}_{\|\mathcal{F}^{\ell}(O) - \mathcal{F}^{\ell}(I)\|^2} + \Gamma \sum_{\ell} \underbrace{\mathcal{L}^{\ell}_{style}}_{\|G[\mathcal{F}^{\ell}(O)] - G[\mathcal{F}^{\ell}(S)]\|^2}$$

Где I, S, O — изображение, стиль и результат соответственно. $\mathcal{F}_{\ell}(\cdot)$ — матрица активаций ℓ-ого слоя, а $G[\cdot]$ — матрица Грама. Γ — гиперпараметр.

[1] В работе «Demystifying Neural Style Transfer» (Li et al.) было показано, что оптимизация разницы матриц Грама эквивалентна минимизации Maximum Mean Discrepancy между распределениями фичей.

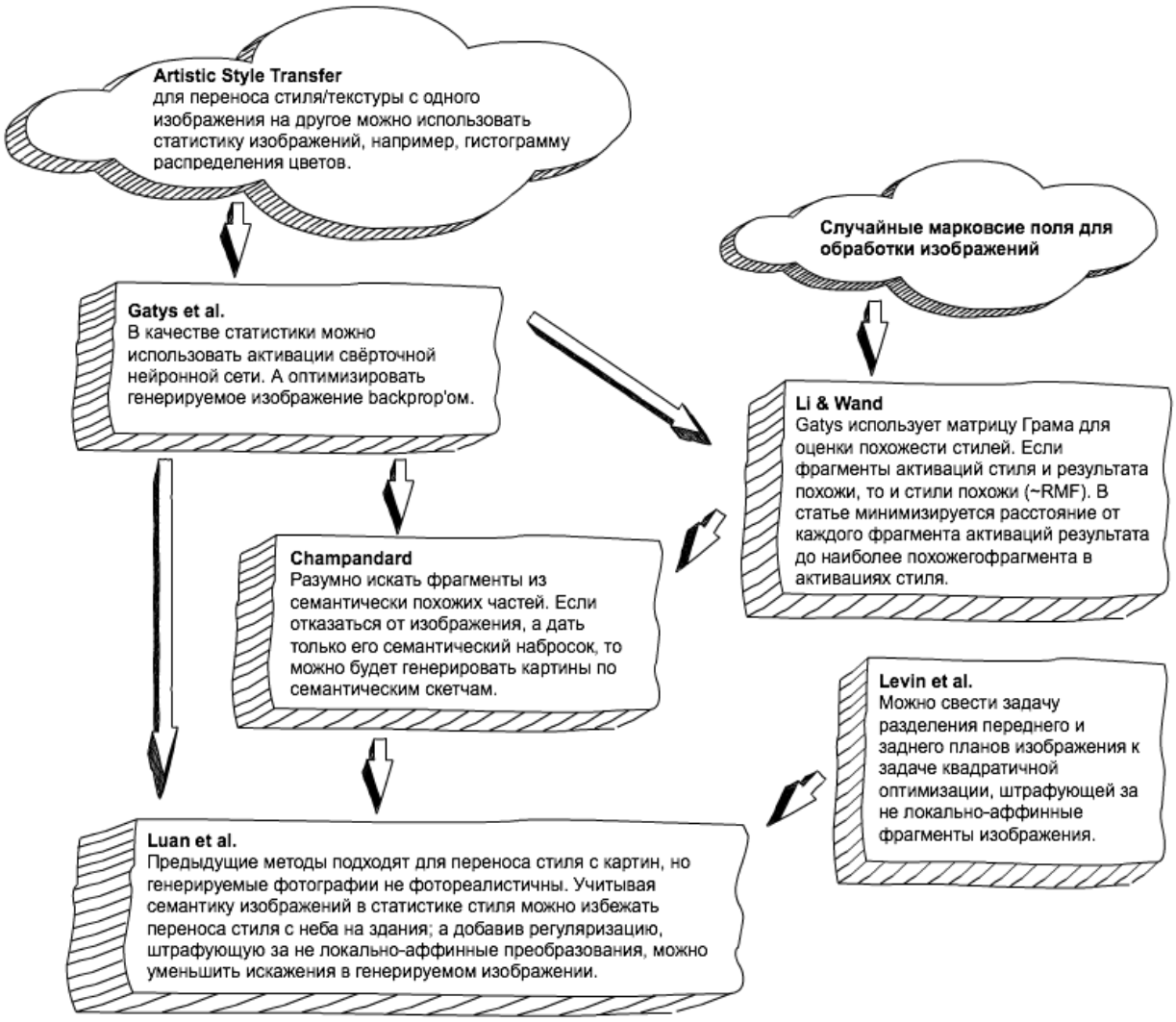

Рис. 1: Диаграмма зависимостей статей.

Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis (Li & Wand)

Если в предыдущий подход к оценке качества стиля основывался на статистики из матриц Грама, то в этой статье предлагают смотреть на небольшие фрагменты матрицы активации и оценивать качество переноса стиля с помощью нахождения наиболее похожего фрагмента в функциях активации S.

$$\mathcal{L}^{\ell}_{style+MRF} = \sum_i \left(\mathcal{F}^{\ell}_i(O) - \mathcal{F}^{\ell}_{NN(i)}(S)\right)^2$$

Где

$$NN(i) = \text{argmin}_{\text{j}} \frac{\mathcal{F}^{\ell}_{\text{i}}(\text{O}) \cdot \mathcal{F}^{\ell}_{\text{j}}(\text{S})}{|\mathcal{F}^{\ell}_{\text{i}}(\text{O})| \cdot |\mathcal{F}^{\ell}_{\text{j}}(\text{S})|}$$

— номер ближайшего в смысле кросс-корреляции фрагмента из функции активации S.

Semantic Style Transfer and Turning Two-Bit Doodles into Fine Artwork (Champandard)

В этой работе было предложено использовать семантическую информацию для генерации изображений по скетчам/дудлам. Основная идея в том, чтобы учитывать семантическую составляющую при подборе подходящего фрагмента. То есть к функции активации добавляется семантическая информация и ближайший фрагмент ищется уже среди таких размеченных фрагментов.

В этой работе нет I, а есть только семантическая разметка, поэтому $\mathcal{L}_{content}$ не учитывается.

A Closed-From Solution to Natural Image Matting (Levin et al.)

Алгоритм разделения переднего и заднего планов изображения, детали которого слишком сложно описать коротко :)

В статье задача разделения фонов сводится к задаче квадратичной оптимизации, наказывающей за не локально-аффинные преобразования. Мы воспользуемся этой идеей и формулировкой оптимизационной задачей, чтобы преобразование I → O было близко к локально-аффинному, чтобы избежать искажений (грубо говоря, кривых окон, шатающихся зданий и т. д.).

Deep Photo Style Transfer (Luan et al.)

Предыдущие подходы отлично могут «нарисовать» фотографию в стиле картины. Но результаты либо не фотореалистичны, либо не переносят стиль в полной мере. Теперь правильно подобрав образец для переноса стиля, можно «изменять» время суток, освещение, погоду или художественную обработку своей фотографии так, чтобы она оставалась «фотографией».

Основные трудности это сохранение структуры изображения и семантически аккуратный перенос стиля. В статье две больших идеи: (1) использовать семантическую сегментацию при подсчёте статистики изображений, (2) использовать регуляризацию, наказывающую не локально-аффинные преобразования I → O.

Напомню, что Gatys et al. используют $\mathcal{L}^{\ell}_{style} = \|G[\mathcal{F}^{\ell}(O)] - G[\mathcal{F}^{\ell}(S)]\|^2$ в качестве стилистической функции потери. Luan et al. определяют $\mathcal{L}^{\ell}_{style+} = \sum_c^C \|G[\mathcal{F}^{\ell}(O)M_{\ell,c}(I)] - G[\mathcal{F}^{\ell,c}(S)M_{\ell,c}(S)]\|^2$, (где $M_{\ell,c}$ — маска, получаемая из семантической сегментации) — отдельно рассматривают статистики по каждому из семантических каналов. Это позволяет избежать переноса нежелательного переноса стиля, например, с океана на небо или здания.

Автоматическое разделение фотореалистичных и нефотореалистичных изображений нерешенная задача. Но в этом случае можно пользоваться тем, что подаваемое на вход изображение I фотореалистично, значит, достаточно лишь сохранить это свойство. Чтобы сохранить фотореалистичность авторы предложили искать локально-аффинное преобразование I → O. С одной стороны, это довольно мягкое условие, потому что у каждого фрагмента изображения может быть своё аффинное преобразование, поэтому облачное небо может изменить свой цвет. С другой стороны, границы / края должны остаться на своём месте, потому что в каждом фрагменте сохраняет присутствие / отсутствие ребра. За не локально-аффинность можно штрафовать с помощью вычисления квадратичной ошибки предложенной Levin et al: $\mathcal{L}_{matting} = \sum_1^3 vec(O)^T \mathcal{M}_I \, vec(O)$, где $vec(\cdot)$ — оператор векторизации, «выпрямляющий» матрицу в вектор, $\mathcal{M}_I$ — matting Laplacian вычисленный для исходного изображения.

Недостаток подхода в том, что он довольно трудоёмкий, но позже были предложены более быстрые, но менее фотореалистичные методы.

Итого:

$$\mathcal{L}_{total} = \sum_\ell \underbrace{\mathcal{L}^\ell_{context}}_{\|\mathcal{F}^\ell(O)-\mathcal{F}^\ell(I)\|^2} + \Gamma \sum_\ell \underbrace{\mathcal{L}^\ell_{style+}}_{\sum_c^C \|G[\mathcal{F}^\ell(O)M_{\ell,c}(I)]-G[\mathcal{F}^\ell(S)M_{\ell,c}(S)]\|^2} + \lambda \underbrace{\mathcal{L}_{matting}}_{\sum_1^3 vec(O)^T \mathcal{M}_I \, vec(O)}$$

Где I, S, O — изображение, стиль и результат соответственно. $\mathcal{F}_\ell(\cdot)$ — матрица активаций ℓ-ого слоя, $M_{\ell,c}$ — маска, получаемая из семантической сегментации, а $G[\cdot]$ — матрица Грама. C — число каналов сегментирования. $vec(\cdot)$ — оператор векторизации, «выпрямляющий» матрицу в вектор, $\mathcal{M}_I$ — matting Laplacian. Γ и λ — гиперпараметры.

Детали подхода и реализации

Оптимизацию рекомендуют делать в два этапа: начать с белого шума, применить к нему NST с улучшенной стилевой функцией потерь, после чего включить фотореалистичную регуляризацию.

Как всегда в задачах переноса стиля используется VGG19[2]. Для content loss используют conv4_2, для style loss — conv1_1, conv2_1, conv3_1, conv4_1 и conv5_1. В качестве нелинейности используется average pooling (рекомендация из статьи Gatys et al.).

Гиперпараметры рекомендуют следующие: $\Gamma = 100$, $\lambda = 10^4$.

[2] https://www.reddit.com/r/MachineLearning/comments/7rrrk3/d_eat_your_vggtables_or_why_does_neural_style/