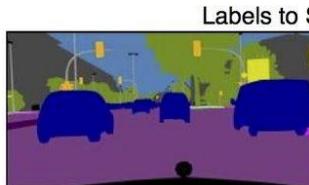


Contrastive Learning for Unpaired Image-to-Image Translation

Шапкин Антон
Сусла Диана
Ким Михаил
Каратеева Екатерина

Image-to-Image translation

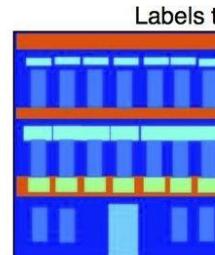
Хотим научиться строить соответствия между входным и выходным изображениями



input



output



input



output

BW to Color



input



output



input



output



input



output

Labels to Facade

input

output

Edges to Photo



input

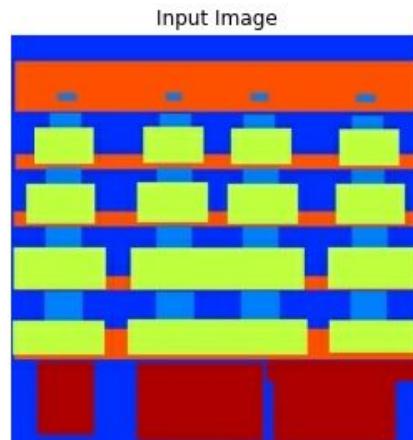


output

Paired Image-to-Image translation

Выборка состоит из пар $(x, y)_i$, существует соответствие между x и y .

Методы решения: Fully-Convolutional, Conditional Adversarial Networks, ...



Unpaired Image-to-Image translation

Есть исходный набор $\{x_i\}$ и целевой $\{y_j\}$, без информации о соответствии друг другу (например, изображения лошадей и зебр)



zebra → horse

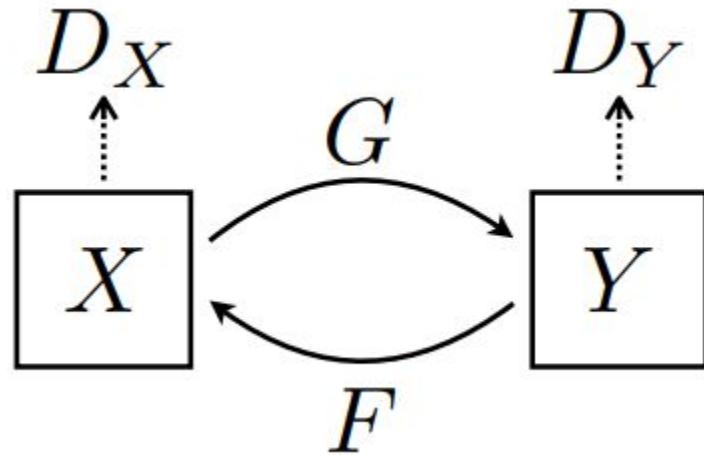


apple → orange



Cycle Consistency Loss

Идея: выучить 2 отображения $G: X \rightarrow Y$ и $F: Y \rightarrow X$. Тогда, $F(G(X)) \approx X$. Введем также дискриминаторы D_X и D_Y .



Cycle Consistency Loss

Идея: выучить 2 отображения $G: X \rightarrow Y$ и $F: Y \rightarrow X$. Тогда, $F(G(X)) \approx X$. Введем также дискриминаторы D_X и D_Y .

Adversarial Loss:

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = & \mathbb{E}_{y \sim p_{\text{data}}(y)} [\log D_Y(y)] \\ & + \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log(1 - D_Y(G(x)))]\end{aligned}$$

Стандартный Adversarial loss, генератор минимизирует данную функцию потерь, в то время как дискриминатор максимизирует

Cycle Consistency Loss

Идея: выучить 2 отображения $G: X \rightarrow Y$ и $F: Y \rightarrow X$. Тогда, $F(G(X)) \approx X$. Введем также дискриминаторы D_X и D_Y .

Adversarial Loss:

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = & \mathbb{E}_{y \sim p_{\text{data}}(y)} [\log D_Y(y)] \\ & + \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log(1 - D_Y(G(x)))]\end{aligned}$$

Cycle Consistency Loss:

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G, F) = & \mathbb{E}_{x \sim p_{\text{data}}(x)} [\|F(G(x)) - x\|_1] \\ & + \mathbb{E}_{y \sim p_{\text{data}}(y)} [\|G(F(y)) - y\|_1].\end{aligned}$$

Cycle Consistency Loss

Adversarial Loss:

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) &= \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] \\ &\quad + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x)))]\end{aligned}$$

Cycle Consistency Loss:

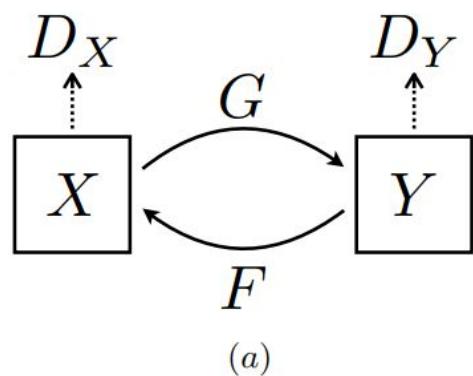
$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G, F) &= \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] \\ &\quad + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1].\end{aligned}$$

Full Objective:

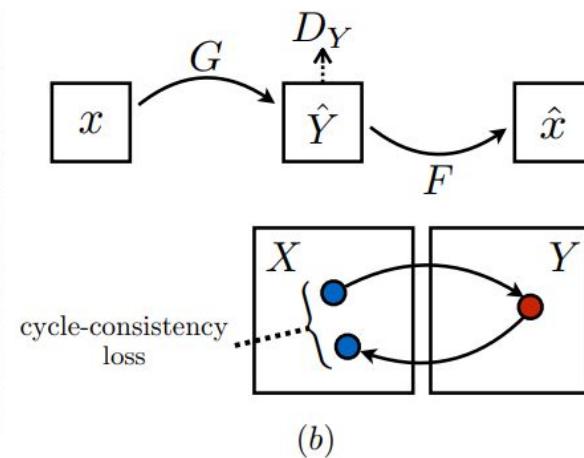
$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) &= \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) \\ &\quad + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) \\ &\quad + \lambda \mathcal{L}_{\text{cyc}}(G, F),\end{aligned}$$

Cycle Consistency Loss

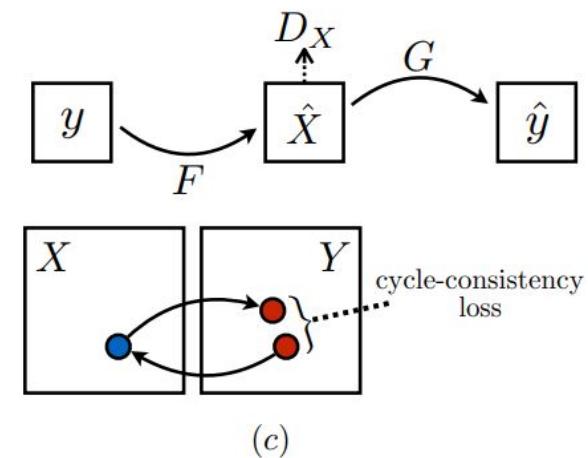
Довольно простой метод, который в свое время (2017 год) превзошел всех конкурентов.



model architecture



forward cycle-consistency loss



backward cycle-consistency loss

GCGAN

Идея: определим функцию трансформации $f: X \rightarrow Z$ (например, вертикальный или горизонтальный переворот).

GCGAN

Идея: определим функцию трансформации $f: X \rightarrow Z$ (например, вертикальный или горизонтальный переворот).

Возьмем $x \in X$ и посчитаем $z = f(x)$. Для пространств X и Z есть генераторы G_x и G_z соответственно.

GCGAN

Идея: определим функцию трансформации $f: X \rightarrow Z$ (например, вертикальный или горизонтальный переворот).

Возьмем $x \in X$ и посчитаем $z = f(x)$. Для пространств X и Z есть генераторы G_x и G_z соответственно.

Посчитаем $y_x = G_x(x)$ и $y_z = G_z(z)$. Тогда, должно выполняться $y_z = f(y_x)$

GCGAN

Идея: определим функцию трансформации $f: X \rightarrow Z$ (например, вертикальный или горизонтальный переворот).

Возьмем $x \in X$ и посчитаем $z = f(x)$. Для пространств X и Z есть генераторы G_X и G_Z соответственно.

Посчитаем $y_X = G_X(x)$ и $y_Z = G_Z(z)$. Тогда, должно выполняться $y_Z \approx f(y_X)$

Geometry consistency loss:

$$\begin{aligned}\mathcal{L}_{geo}(G_{XY}, G_{\tilde{X}\tilde{Y}}, X, Y) &= \mathbb{E}_{x \sim P_X} [\|G_{XY}(x) - f^{-1}(G_{\tilde{X}\tilde{Y}}(f(x)))\|_1] \\ &\quad + \mathbb{E}_{x \sim P_X} [\|G_{\tilde{X}\tilde{Y}}(f(x)) - f(G_{XY}(x))\|_1].\end{aligned}$$

GCGAN

Geometry consistency loss:

$$\begin{aligned}\mathcal{L}_{geo}(G_{XY}, G_{\tilde{X}\tilde{Y}}, X, Y) = & \mathbb{E}_{x \sim P_X} [\|G_{XY}(x) - f^{-1}(G_{\tilde{X}\tilde{Y}}(f(x)))\|_1] \\ & + \mathbb{E}_{x \sim P_X} [\|G_{\tilde{X}\tilde{Y}}(f(x)) - f(G_{XY}(x))\|_1].\end{aligned}$$

Full objective:

$$\begin{aligned}\mathcal{L}_{GcGAN}(G_{XY}, G_{\tilde{X}\tilde{Y}}, D_Y, D_{\tilde{Y}}, X, Y) = & \mathcal{L}_{gan}(G_{XY}, D_Y, X, Y) \\ & + \mathcal{L}_{gan}(G_{\tilde{X}\tilde{Y}}, D_{\tilde{Y}}, X, Y) \\ & + \lambda \mathcal{L}_{geo}(G_{XY}, G_{\tilde{X}\tilde{Y}}, X, Y)\end{aligned}$$

Patchwise Contrastive Learning

Метод, в отличии от GCGAN и CycleGAN не требует вспомогательных генераторов и дискриминаторов, учит отображение в одном направлении.

Patchwise Contrastive Learning

Метод, в отличии от GCGAN и CycleGAN не требует вспомогательных генераторов и дискриминаторов, учит отображение в одном направлении.

Пусть имеются генератор G , состоящий из G_{enc} и G_{dec} :

$$G(x) = G_{\text{dec}}(G_{\text{enc}}(x))$$

дискриминатор D , набор исходных изображений X и набор целевых изображений Y .

Adversarial loss:

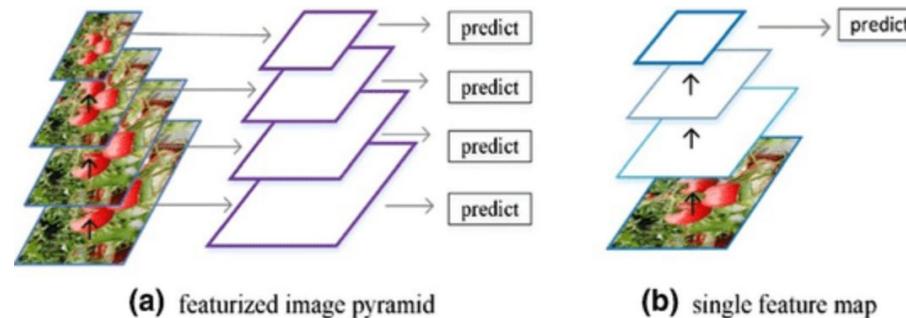
$$\mathcal{L}_{\text{GAN}}(G, D, X, Y) = \mathbb{E}_{y \sim Y} \log D(y) + \mathbb{E}_{x \sim X} \log(1 - D(G(x)))$$

Patchwise Contrastive Learning

Идея: Patchwise Contrastive Loss

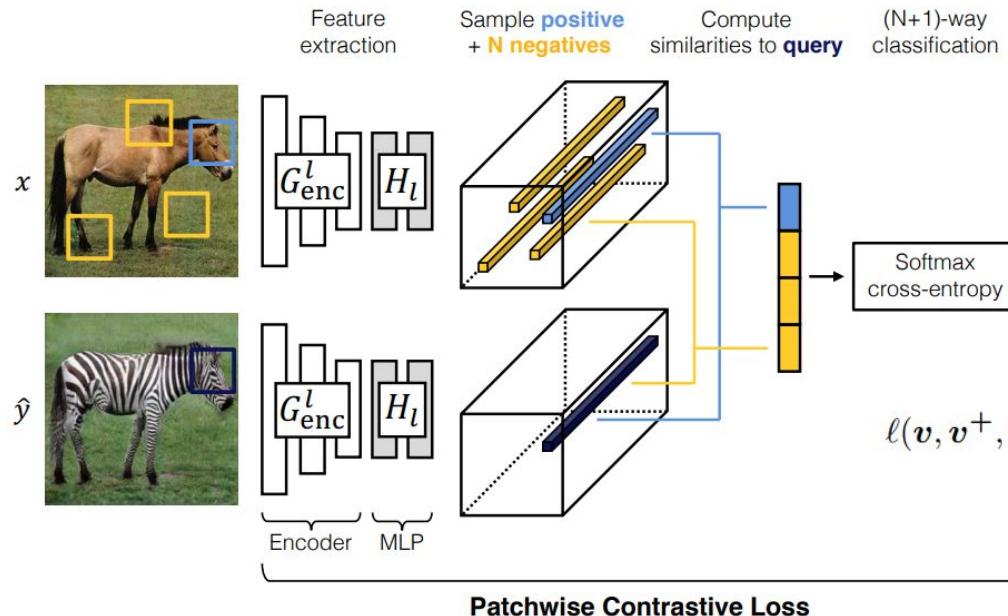
Важно, чтобы на входной и предсказанной картинках контент был одинаковым не только на уровне целого изображения, но и на уровне патчей.

Каждая feature map какого-то слоя G_{enc} соответствует входному изображению в том или ином разрешении.



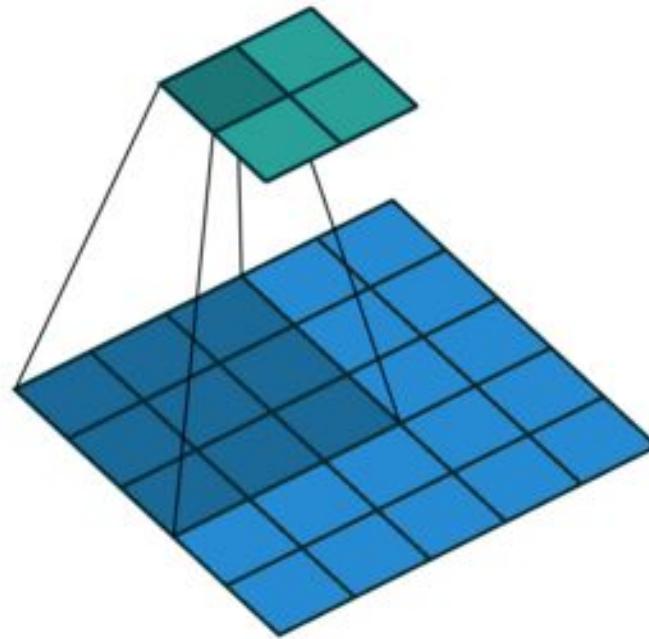
Patchwise Contrastive Learning

Идея: Patchwise Contrastive Loss



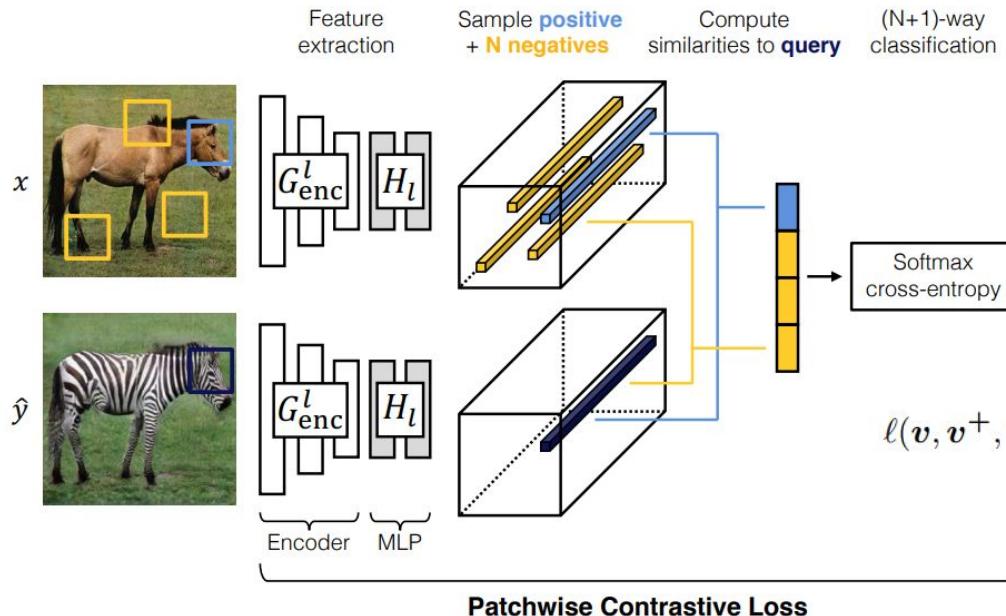
$$\ell(\mathbf{v}, \mathbf{v}^+, \mathbf{v}^-) = -\log \left[\frac{\exp(\mathbf{v} \cdot \mathbf{v}^+ / \tau)}{\exp(\mathbf{v} \cdot \mathbf{v}^+ / \tau) + \sum_{n=1}^N \exp(\mathbf{v} \cdot \mathbf{v}_n^- / \tau)} \right]$$

Patchwise Contrastive Learning



Patchwise Contrastive Learning

Идея: Patchwise Contrastive Loss



$$\ell(\mathbf{v}, \mathbf{v}^+, \mathbf{v}^-) = -\log \left[\frac{\exp(\mathbf{v} \cdot \mathbf{v}^+ / \tau)}{\exp(\mathbf{v} \cdot \mathbf{v}^+ / \tau) + \sum_{n=1}^N \exp(\mathbf{v} \cdot \mathbf{v}_n^- / \tau)} \right]$$

Patchwise Contrastive Learning

Идея: Patchwise Contrastive Loss

Выберем L слоев G_{enc} и пропустим их через 2-слойную MLP сеть H_l , получив набор признаков $\{z_l\}_L = \{H_l(G_{\text{enc}}^l(\mathbf{x}))\}_L$, где G_{enc}^l - выход l-ого выбранного слоя.

Patchwise Contrastive Learning

Идея: Patchwise Contrastive Loss

Выберем L слоев (G_{enc}) и пропустим их через 2-слойную MLP сеть H_l , получив набор признаков $\{z_l\}_L = \{H_l(G_{\text{enc}}^l(\mathbf{x}))\}_L$, где G_{enc}^l - выход l -ого выбранного слоя.

Пусть $s \in \{1, \dots, S_i\}$ номер патча признаковой карты i -го слоя, S_i - число патчей на i -ом слое.

Тогда $z_i \in \mathbb{R}^{S_i \times C_i}$, где C_i - количество выходных каналов на i -ом слое.

Patchwise Contrastive Learning

Идея: Patchwise Contrastive Loss

Получим набор признаков $\{z_l\}_L = \{H_l(G_{\text{enc}}^l(\mathbf{x}))\}_L$, где G_{enc}^l - выход l-ого выбранного слоя.

Пусть $s \in \{1, \dots, S_i\}$ номер патча признаковой карты i-го слоя, S_i - число патчей на i-ом слое.

Тогда $z_s \in \mathbb{R}^{S_i \times C_i}$, где C_i - количество выходных каналов на i-ом слое.

Аналогично сделаем для предсказанного изображения и получим набор $\{\hat{z}_l\}_L = \{H_l(G_{\text{enc}}^l(G(\mathbf{x})))\}_L$

Patchwise Contrastive Learning

Получим набор признаков $\{\mathbf{z}_l\}_L = \{H_l(G_{\text{enc}}^l(\mathbf{x}))\}_L$, где G_{enc}^l - выход l-ого выбранного слоя.

Пусть $s \in \{1, \dots, S_i\}$ номер патча признаковой карты i-го слоя, S_i - число патчей на i-ом слое.

Аналогично сделаем для предсказанного изображения и получим набор $\{\hat{\mathbf{z}}_l\}_L = \{H_l(G_{\text{enc}}^l(G(\mathbf{x})))\}_L$

PatchNCE Loss:

$$\mathcal{L}_{\text{PatchNCE}}(G, H, X) = \mathbb{E}_{\mathbf{x} \sim X} \sum_{l=1}^L \sum_{s=1}^{S_l} \ell(\hat{\mathbf{z}}_l^s, \mathbf{z}_l^s, \mathbf{z}_l^{S_l \setminus s})$$

Patchwise Contrastive Learning

PatchNCE Loss: $\mathcal{L}_{\text{PatchNCE}}(G, H, X) = \mathbb{E}_{\mathbf{x} \sim X} \sum_{l=1}^L \sum_{s=1}^{S_l} \ell(\hat{\mathbf{z}}_l^s, \mathbf{z}_l^s, \mathbf{z}_l^{S \setminus s})$

Full objective:

$$\mathcal{L}_{\text{GAN}}(G, D, X, Y) + \lambda_X \mathcal{L}_{\text{PatchNCE}}(G, H, X) + \lambda_Y \mathcal{L}_{\text{PatchNCE}}(G, H, Y)$$

CUT: $\lambda_X = \lambda_Y = 1$

FastCUT: $\lambda_X = 20, \lambda_Y = 0$

Patchwise Contrastive Learning

PatchNCE Loss: $\mathcal{L}_{\text{PatchNCE}}(G, H, X) = \mathbb{E}_{x \sim X} \sum_{l=1}^L \sum_{s=1}^{S_l} \ell(\hat{z}_l^s, z_l^s, z_l^{S \setminus s})$

Full objective:

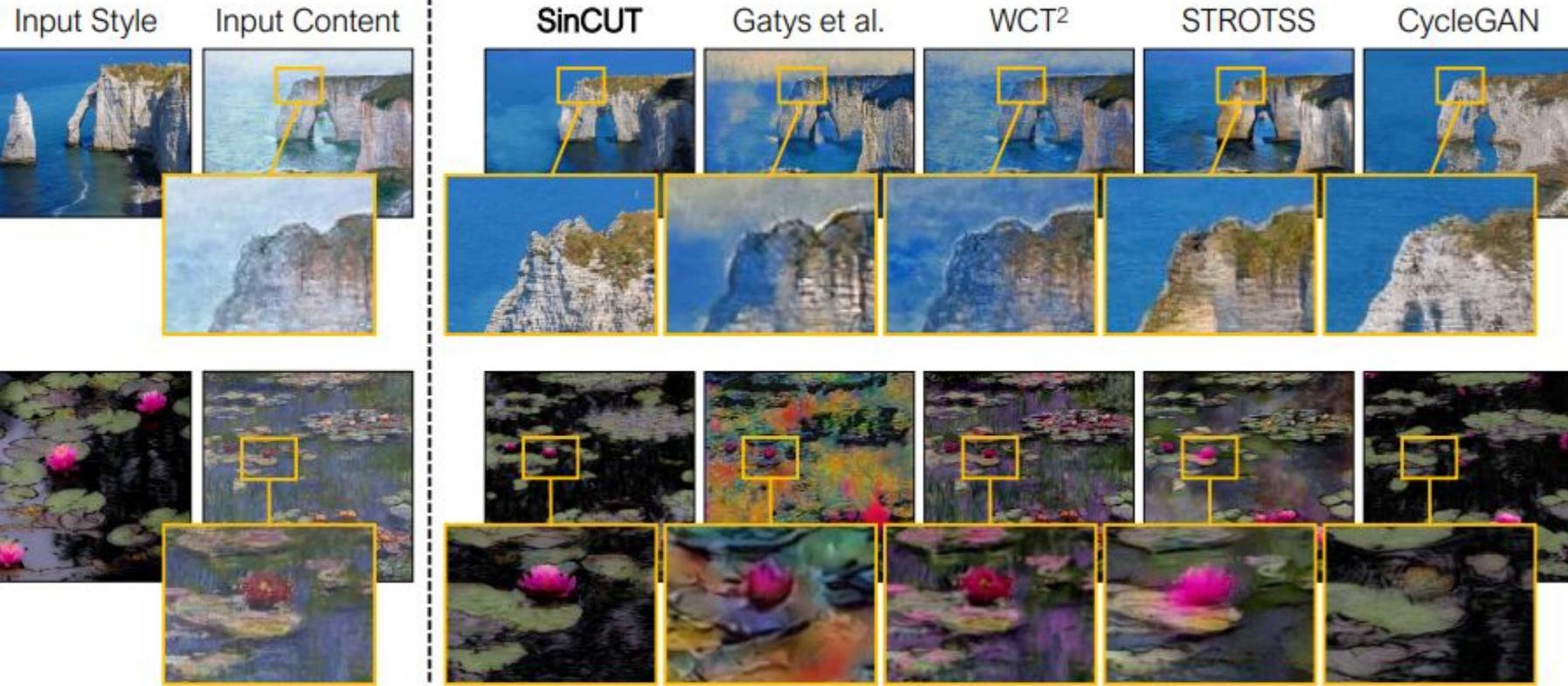
$$\mathcal{L}_{\text{GAN}}(G, D, X, Y) + \lambda_X \mathcal{L}_{\text{PatchNCE}}(G, H, X) + \lambda_Y \mathcal{L}_{\text{PatchNCE}}(G, H, Y)$$

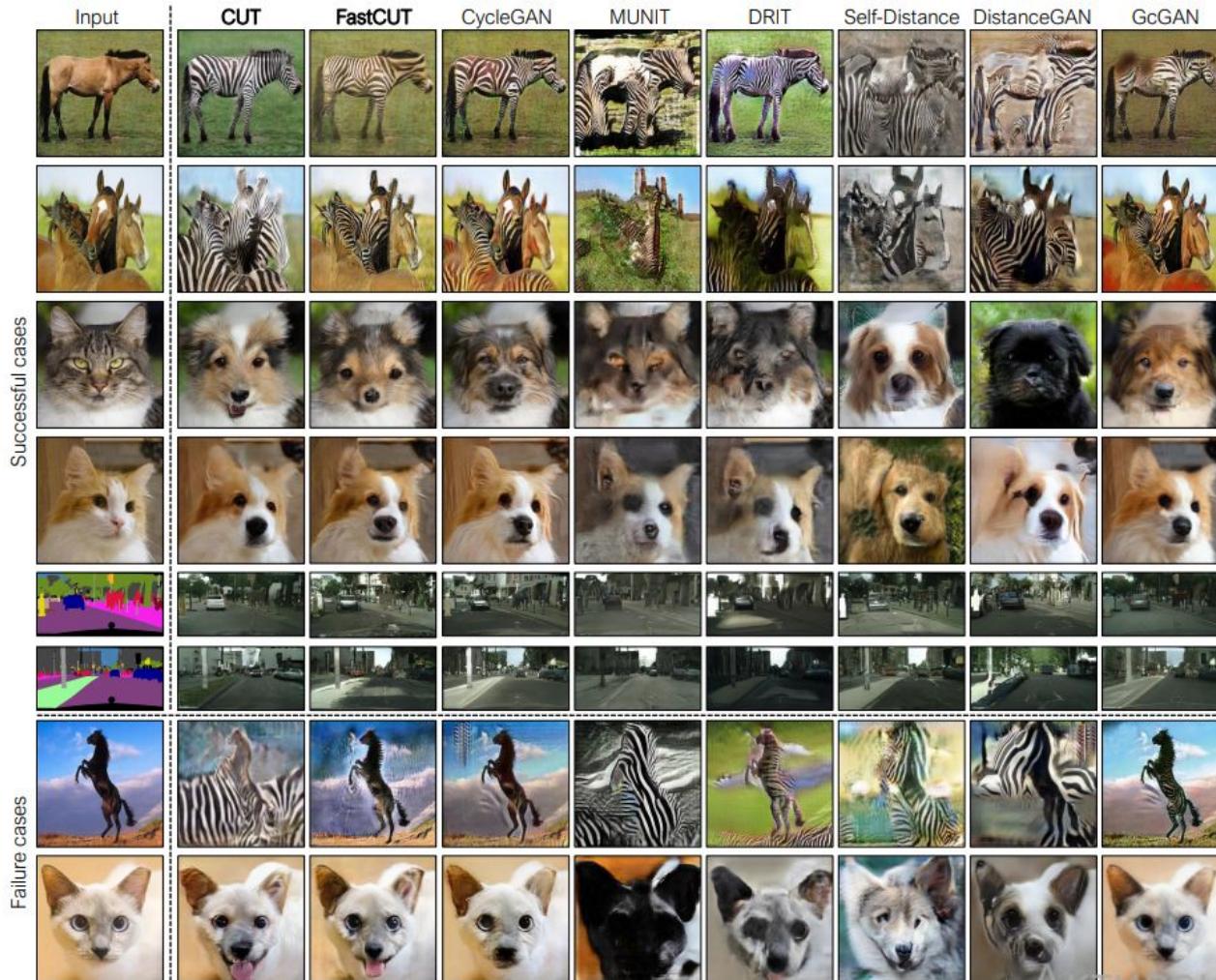
CUT: $\lambda_X = \lambda_Y = 1$

FastCUT: $\lambda_X = 20, \lambda_Y = 0$

SinCUT: работает, даже если в данных всего 1 картинка

SinCUT





Результаты

Method	Cityscapes				Cat→Dog	Horse→Zebra		
	mAP↑	pixAcc↑	classAcc↑	FID↓	FID↓	FID↓	sec/iter↓	Mem(GB)↓
CycleGAN [89]	20.4	55.9	25.4	76.3	85.9	77.2	0.40	4.81
MUNIT [44]	16.9	56.5	22.5	91.4	104.4	133.8	0.39	3.84
DRIT [41]	17.0	58.7	22.2	155.3	123.4	140.0	0.70	4.85
Distance [4]	8.4	42.2	12.6	81.8	155.3	72.0	0.15	2.72
SelfDistance [4]	15.3	56.9	20.6	78.8	144.4	80.8	0.16	2.72
GCGAN [18]	21.2	63.2	26.6	105.2	96.6	86.7	0.26	2.67
CUT	24.7	68.8	30.7	56.4	76.2	45.5	0.24	3.33
FastCUT	19.1	59.9	24.3	68.8	94.0	73.4	0.15	2.25

Результаты

Method	Training settings					Testing datasets			
	Id	Negs	Layers	Int	Ext	Horse → Zebra		Cityscapes	
						FID↓	mAP↑	FID↓	mAP↑
CUT (default)	✓	255	All	✓	✗	45.5	56.4	24.7	
no id	✗	255	All	✓	✗	39.3	68.5	22.0	
no id, 15 neg	✗	15	All	✓	✗	44.1	59.7	23.1	
no id, 15 neg, last	✗	15	Last	✓	✗	38.1	114.1	16.0	
last	✓	255	Last	✓	✗	441.7	141.1	14.9	
int and ext	✓	255	All	✓	✓	56.4	64.4	20.0	
ext only	✓	255	All	✗	✓	53.0	110.3	16.5	
ext only, last	✓	255	Last	✗	✓	60.1	389.1	5.6	

Id - identity loss, Negs - num of negatives, Layers - using only the last layer of the encoder or all layers, Int and Ext - where patches are sampled (from X and Y)

Рецензия

Содержание работы:

- Решается задача непарного перевода изображений
- Метод основан на максимизации взаимной информации между патчами входа и выхода, используя структуру, основанную на contrastive learning
- Используется Patchwise Contrastive Learning

Актуальность:

- Популярность contrastive learning
- Актуальность решения задачи перевода изображений: перенос стиля и т.д.

Сильные стороны

- Сама статья написана очень подробно и довольно понятно. Пояснены все ключевые моменты - от функций потерь и до выбора патчей входного изображения + примеры
- Модель довольно проста, относительно других работ по данной теме, которые используют большее число функций потерь и гиперпараметры. Модель использует метод Patchwise Contrastive Learning, который в отличии от GCGAN и CycleGAN не требует вспомогательных генераторов и дискриминаторов, учит отображение в одном направлении
- До данной работы не было алгоритмов перевода изображений, не использующих какую-либо заранее определенную функцию сходства (L_1 или perceptual loss)
- В статье есть сравнение с основными SOTA моделями. По всем рассматриваемым метрикам CUT лучше других моделей. FastCUT сокращает потребление памяти в сравнении с другими SOTA моделями
- Предложенный метод может быть расширен до случая, когда на входе и выходе модели всего одно изображение (SimCUT)
- Большой ссылочный аппарат (91 работа)

Слабые стороны

- Есть момент с тем, что неясно, зачем утверждается, что для контрастирования берутся L слоёв сети, если в итоге эксперименты проводятся либо для всех, либо для последнего
- Не пояснено, как именно патчи выбираются из изображения. Случайно, нет?
- Еще пару вопросов к коду

Воспроизводимость + оценка по критериям НИПСа

Воспроизводимость: есть open source код модели. Также в самой статье приводятся пояснения, благодаря чему результаты статьи воспроизводимы

Оценка: 9

Уверенность: 4

Контекст: авторы

Работа написана в 2020 году, была принята на конференцию ECCV

Авторы статьи:

- **Taesung Park** – Adobe Research, в этом году защитил диссертацию, один из авторов CycleGAN, много статей, связанных с генерацией изображений, редактированием изображений с помощью DL
- **Alexei A. Efros** – профессор в Бёркли, с 1998 года занимается CV, есть несколько работ по Image-to-Image translation и по Contrastive Learning
- **Richard Zhang** – Adobe Research, в основном статьи по CV – редактирование изображений/генерация, автор статьи про раскраску чёрно-белых фото, есть несколько по 3D CV
- **Jun-Yan Zhu** – доцент Университета Карнеги–Меллона, много работ по Image-To-Image translation и по GAN вообще, в том числе один из авторов GAN Dissection

Контекст: предпосылки

- Популярность Contrastive Learning
- Попытки заменить Cycle-Consistency обучением кросс-доменной инвариантностью

Работы, связанные с Contrastive Learning:

- Стандартная функция потерь для Contrastive Learning – [InfoNCE](#)
- [Representation Learning with Contrastive Predictive Coding](#) – contrastive learning между патчами: позитивные пары – представления патча и представления этого же патча, предсказанного по патчам, которые на изображении выше текущего, негативные пары составляют представления других патчей (с этого же изображения и со всей выборки)

Контекст: предпосылки

Работы, связанные с заменой Cycle-Consistency:

- [GcGAN](#): при переводе в другой домен должны сохраняться геометрические свойства, так как контент изображения при таких трансформациях не меняется.
- [DistanceGAN](#): перевод в другой домен должен сохранять расстояние между изображениями: расстояние между двумя изображениями из X должно быть близко к расстоянию между соответствующими сгенерированными изображения

Контекст: связь с другими исследованиями

На данный момент 124 цитирований, в основном цитируют из-за связи с Contrastive Learning:

- [Dual Contrastive Loss and Attention for GANs](#) – переформулировка классической функции потерь для обучения GAN с точки зрения Contrastive Learning: дискриминатор учится отличать одну реальную картинку от батча сгенерированных и наоборот
- [Cross-Modal Contrastive Learning for Text-to-Image Generation](#) – GAN для Text-to-Image, используют Contrastive Learning везде: между сгенерированными изображениями и реальными, между представлениями текста и представлением соответствующего изображения, между представлениями слов и патчей изображения

Контекст: связь с другими исследованиями

Что можно почитать, если статья показалась интересной:

- [Instance-wise Hard Negative Example Generation for Contrastive Learning in Unpaired Image-to-Image Translation](#) – улучшение CUT путём генерации сложных негативных примеров
- [Exploring Cross-Image Pixel Contrast for Semantic Segmentation](#) – как с помощью contrastive learning на уровне патчей и пикселей улучшить качество решения задачи семантической сегментации
- [Cntr-GAN](#) – как с помощью contrastive learning и аугментаций получить ванильным GAN'ом качество, сравнимое с SOTA
- [ContraGAN](#) – использование contrastive learning при обучении Conditional GAN

Контекст: идеи и гипотезы

- В статье предложен Multilayer Contrastive Learning подход, ранее никто не контрастировал выходы с промежуточных слоев
- В Ablation Study исследованы две крайности: использование выходов со всех слоев, либо только с последнего
- В качестве adversarial loss используется классический вариант

Контекст: практическое применение

- Генерация обучающих данных: разметка -> данные и наоборот
- Доменная адаптация
- Style Transfer

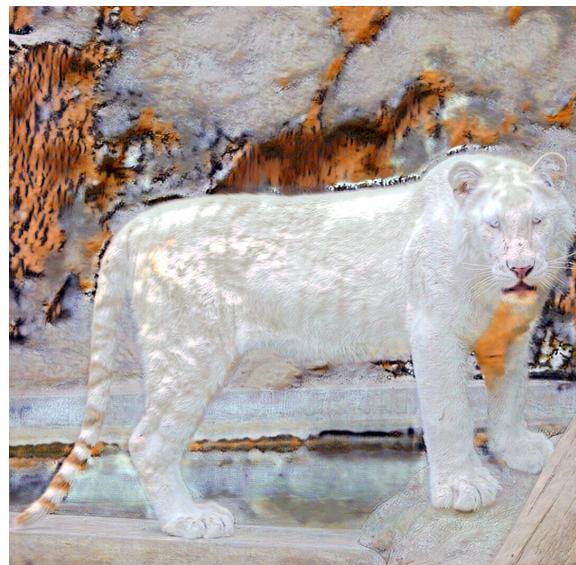
Эксперименты

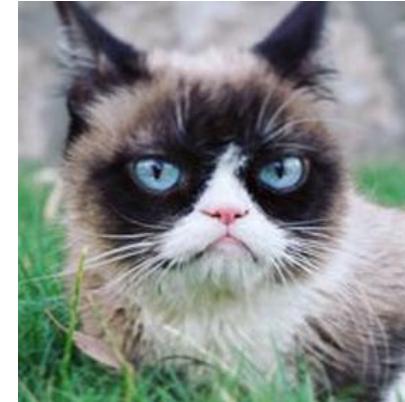


Эксперименты



Эксперименты





Input



Output



Input



Output



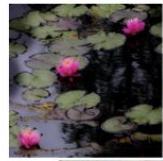
Russian blue cat → Grumpy cat



Input



SinCUT



Input



SinCUT



Input



SinCUT



Input



SinCUT



1ый слой: 186



2ой слой: 228



3 слой: 195



1, 2 слой: 195



2, 3 слой: 207



1, 3 слой: 202



все три: 180

