

А.В. Аверченков
А.А. Андросов
Ю.А. Малахов

Анализ и применение генеративно-сопоставительных сетей для получения изображений высокого качества

В статье рассматриваются генеративно-сопоставительные сети для получения изображений высокого качества. Представлены модели, архитектура и сравнение работы сетей. Рассмотрены особенности построения моделей глубокого обучения в процессе выполнения задачи суперразрешения, а также методы связанные с улучшением производительности.

Ключевые слова: нейронная сеть, высокое разрешение, генерирование, глубокое обучение, изображения.

A.V. Averchenkov
A.A. Androsov
Y.A. Malakhov

Analysis and application of generative-adversarial networks for producing high quality images

The article discusses generative adversarial networks for obtaining high quality images. Models, architecture and comparison of network operation are presented. The features of building deep learning models in the process of performing the super-resolution task, as well as methods associated with improving performance, are considered.

Keywords: neuralnetwork, highresolution, generation, deeplearning, images.

Генеративные сопоставительные нейронные сети (generative adversarial network, GAN) – это модель порождения данных, фактически построенная на основе двух моделей, состоящих между собой в своеобразной игре. Этими двумя «игроками» является генератор и дискриминатор (Ч. Аггарвал [1], С. Николленко, А. Кадури, Е. Архангельская [3], G. Voh, G. Jenkins [6], A. Tsantekidisi др. [10]).

Как правило генеративно-сопоставительные сети используют для генерирования объектов изображения с различными типами контекста. В настоящее время наиболее распространенной сферой применения GAN является обработка изображений. Сеть-генератор предназначена для работы в этих условиях, называют сетью обращения свертки, или деконволюционной сетью (deconvolution network). Наиболее популярные способы создания деконволюционных сетей для GAN обсуждают-

ся в целом ряде работ [4, 5, 8, 11, 12 и др.].

Спектр использования GAN достаточно широк [1, 4, 7, 8, 9, и др.] в данной статье рассмотрены модели GAN глубокого обучения направленные на получение изображений с высоким качеством разрешения изображений.

Чтобы понять GAN, необходимо знать, как работают генеративные алгоритмы, и для этого полезно сравнивать их с дискриминационными алгоритмами. Дискриминационные алгоритмы пытаются классифицировать входные данные, то есть, учитывая особенности экземпляра данных, они предсказывают метку или категорию, к которой относятся эти данные.

Например, учитывая все слова в электронном письме (экземпляр данных), дискриминационный алгоритм может предсказать, является ли сообщение спамом или не спамом. Спам – это одна из меток, а пакет слов, со-

бренных из электронного письма, - это функции, которые составляют входные данные. Когда эта проблема выражена математически, метка называется y , а функции - x . Формулировка $p(y | x)$ используется для обозначения «вероятности y при заданном x », что в этом случае будет означать «вероятность того, что электронное письмо является спамом с учетом содержащихся в нем слов».

Таким образом, дискриминационные алгоритмы отображают характеристики на метки. Они касаются исключительно этой взаимосвязи. Один из способов понять генеративные алгоритмы состоит в том, что они делают противоположное. Вместо того, чтобы предсказывать метку на основе определенных функций, они пытаются предсказать особенности на основе определенной метки [2].

Вопрос, на который пытаются сгенерировать генеративный алгоритм, заключается в следующем: если предположить, что это письмо является спамом, насколько вероятны эти функции? В то время как дискриминационные модели заботятся о связи между y и x , генеративные модели заботятся о том, «как получается x ». Они позволяют захватить $p(x | y)$, вероятность x , заданную y , или вероятность признаков, заданных меткой или категорией. (Тем не менее, генеративные алгоритмы также могут использоваться в качестве классификаторов. Просто так получается, что они могут делать больше, чем классифицировать входные данные.)

Следует отличать дискриминационные модели от генеративных:

- Дискриминационные модели учат границу между классами.
- Генеративные модели моделируют распределение отдельных классов.

Одна нейронная сеть, называемая генератором, генерирует новые экземпляры данных, а другая, дискриминатор, оценивает их на подлинность; то есть дискриминатор решает, принадлежит ли каждый экземпляр данных, которые он просматривает, к фактическому набору данных или нет.

Например сгенерировать рукописные цифры, подобные тем, которые находятся в наборе данных MNIST, взятом из реального мира. Цель дискриминатора, когда показан экземпляр из истинного набора данных MNIST - распознать подлинные.

Тем временем генератор создает новые синтетические изображения, которые он передает дискриминатору. Это делается в надежде, что они тоже будут считаться подлинными,

даже если они являются поддельными. Цель генератора - генерировать проходимые рукописные цифры: лгать, не быть пойманным. Цель дискриминатора состоит в том, чтобы идентифицировать изображения, поступающие от генератора, как поддельные. [1]

Шаги предпринимаемые GAN:

Генератор принимает случайные числа и возвращает изображение.

Это сгенерированное изображение подается в дискриминатор вместе с потоком изображений, взятых из фактического набора данных с истинным основанием.

Дискриминатор принимает как реальные, так и поддельные изображения и возвращает вероятности, числа от 0 до 1, где 1 представляет прогноз подлинности, а 0 - поддельный.

Итак, есть двойной цикл обратной связи:

Дискриминатор находится в петле обратной связи с основополагающей правдой изображений, которые мы знаем.

Генератор находится в контуре обратной связи с дискриминатором.

На рисунке 1 представлена принципиальная схема GAN.

Можно представить GAN как противодействие фальшивомонетчику и полицейскому в игре в кошки-мышки, где фальшивомонетчик учится передавать фальшивые заметки, а полицейский учится их обнаруживать. Оба являются динамическими; то есть, полицейский тоже обучается (для продолжения аналогии, может быть, центральный банк отмечает пропущенные векселя), и каждая сторона постоянно изучает методы другой.

Для MNIST сеть дискриминатора представляет собой стандартную сверточную сеть, которая может классифицировать поступающие в нее изображения, а биномиальный классификатор маркирует изображения как реальные или поддельные. Генератор представляет собой инверсную сверточную сеть, в некотором смысле: в то время как стандартный сверточный классификатор берет изображение и уменьшает частоту его дискретизации для получения вероятности, генератор берет вектор случайного шума и увеличивает частоту дискретизации его до изображения. Первый отбрасывает данные с помощью методов понижающей дискретизации, таких как max-pooling, а второй генерирует новые данные [7].

Max-pooling слоев - это «уменьшение». Оно позволяет более поздним сверточным слоям работать на больших участках данных, потому что небольшие патчи после слоя объединения соответствует гораздо большему патчу перед

ним. Они также делают нас инвариантными к некоторым очень небольшим преобразовани-

ям данных.

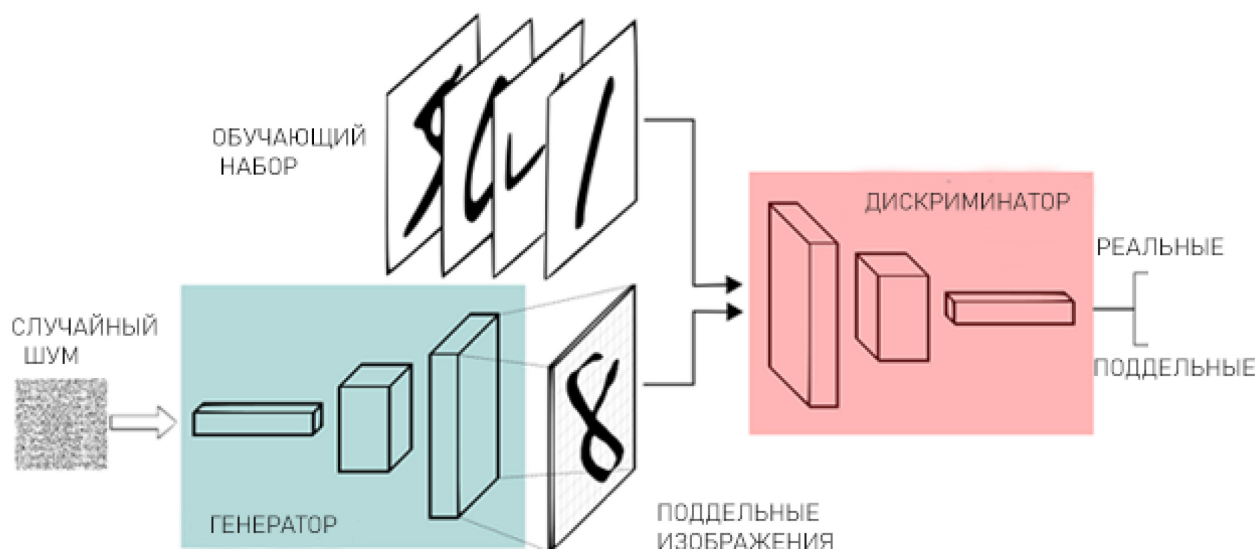


Рис.1. Принципиальная схема генеративно-состязательной сети

Обе сети пытаются оптимизировать другую и противоположную целевую функцию или функцию потерь в игре с нулевым размахом. По сути, это модель актер-критик. Как дискриминатор меняет свое поведение, так и генератор, и наоборот. Их потери давят друг на друга.

Когда обучается дискриминатор, значения генератора держатся постоянными и когда тренируется генератор, постоянным держится дискриминатор. Каждый должен тренироваться против статического противника. Например, это дает генератору лучшее представление о градиенте, которому он должен научиться.

Аналогичным образом, предварительная подготовка дискриминатора к MNIST перед началом обучения генератора установит более четкий градиент.

Каждая сторона GAN может одолеть другую. Если дискриминатор слишком хорош, он будет возвращать значения настолько близкие к 0 или 1, что генератор будет пытаться прочитать градиент. Если генератор слишком хорош, он будет постоянно использовать слабости в дискриминаторе, которые приводят к ложным негативам. Это может быть смягчено соответствующими темпами обучения сетей. Две нейронные сети должны иметь одинаковый «уровень квалификации» [8].

Генеративно-состязательные сети занимают много времени на обучение. На одном графическом процессоре GAN может занять несколько часов, а на одном процессоре - более

суток. Несмотря на сложность настройки и использования, GAN стимулировали множество интересных исследований и написания.

Область применения генеративно-состязательных нейросетей

Искусственные нейронные сети доказали свою эффективность в решении многих задач, однако эта эффективность в большой мере зависит от умения подбирать правильную архитектуру сети под каждую конкретную задачу анализа данных.

GAN имеет огромный потенциал применения. Например, создание картинок для интернет-магазина, аватаров для игр, видеоклипов, сгенерированных автоматически, исходя из музыкального бита произведения, или даже виртуальных ведущих для ТВ-программ. Благодаря работе GAN и генеративных моделей возникает синтез данных, на которых потом будут обучаться другие системы. Этот подход уже используется в современных смартфонах и некоторых программах. Он позволяет менять на фотографии выражение лица, количество морщин и причёску, менять день на ночь, состаривать изображения и т.п.

Генерация потрясающе реалистичных изображений знаменитостей (а на самом деле – несуществующих людей) с помощью PG GAN от nVidia. С помощью этой сети можно также генерировать изображения любых других категорий. Перенос стиля из одного изображе-

ния на другое позволяет при помощи нейросетей делать такие впечатляющие вещи как “превратить лошадь в зебру”.

Изменение эмоций, возраста, выражения лица человека – всего этого можно достичь, правильно обучив и запрограммировав нейронные сети GAN. На практике это выглядит следующим образом: на входе модели подается исходное фото и указывается эмоция, которую нужно показать на выходе.

Также нейронные сети GAN используют для генерации реалистичного видео городской

среды. Например, при создании фильмов, игр, виртуальной реальности.

Основная идея – автоматизация производства контента, который раньше требовал усилий дизайнеров.

Пространство возможных нейронных сетей (рис. 2), в котором ведется поиск, огромно. При проектировании нейронных сетей специалисты пользуются определенными эвристическими правилами и инструментами диагностики, однако полноценной методологией такой поиск назвать сложно, это скорее творческий процесс [1].

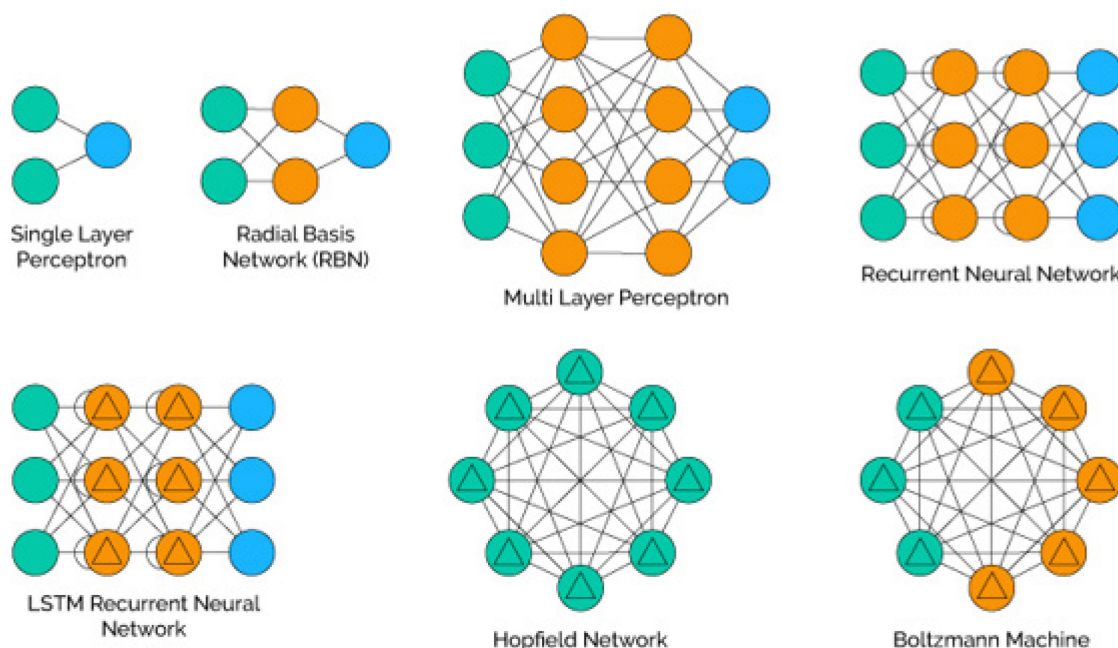


Рис.2. Примеры архитектуры нейросетей

Супер-разрешение — это результат восстановления изображения с высоким разрешением (HR) из данного изображения с низким разрешением (LR). В статье приведены основные подходы для решения задачи генерации супер-разрешения изображения с учителем. Изображение может иметь «более низкое разрешение» из-за меньшего пространственного разрешения (то есть размера) или из-за ухудшения качества (такого как размытие). Мы можем связать изображения HR и LR с помощью следующего уравнения: $LR = degradation(HR)$.

Очевидно, что при применении функции ухудшения качества мы получаем изображение LR из изображения HR. Но можно сделать обратное. Если мы знаем точную функцию деградации, применяя ее инверсию к изображению LR, мы можем восстановить изображе-

ние HR.

Но в этом и заключается проблема. Обычно мы не знаем эту функцию заранее. Непосредственная оценка обратной функции деградации является некорректной задачей. Несмотря на это, методы глубокого обучения доказали свою эффективность для супер-разрешения. Используя изображение HR в качестве цели и изображение LR в качестве входных данных, можно рассматривать это как задачу обучения с учителем.

Одним из простых способов получения данных LR является ухудшение данных HR. Это часто делается с помощью размытия или добавления шума. Изображения с более низким пространственным разрешением также можно масштабировать с помощью классического метода повышения дискретизации, такого как билинейная или бикубическая интерпо-

ляция. Особенности JPEG и аномалии квантования также могут быть использованы для ухудшения изображения.

Помимо классических 2D-сверток, в сетях можно использовать несколько интересных вариантов для улучшения результатов (рис. 3).

Развернутые (злокачественные) свертки могут обеспечить более широкое поле зрения, то

есть, использовать информацию, расположенную на большом расстоянии.

Такие приемы, как пропуск соединений, пространственный пирамидальный пулинг и полносвязные блоки объединяют признаки как низкого, так и высокого уровня для повышения производительности.

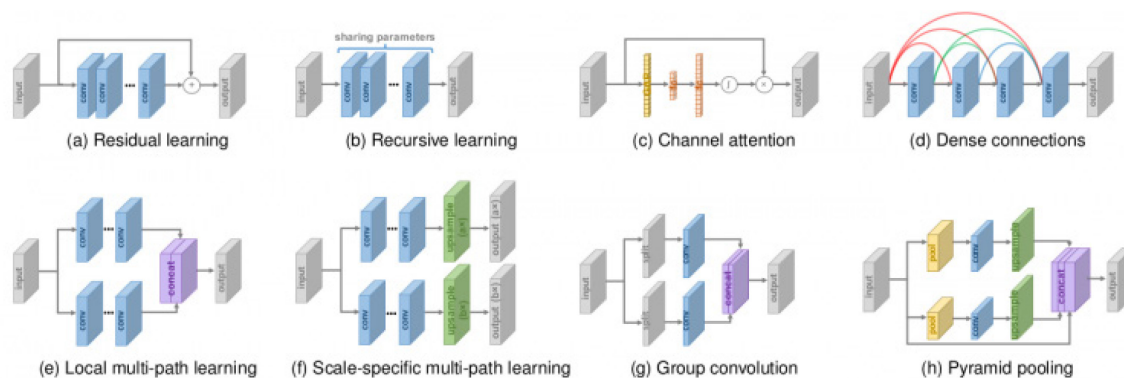


Рис.3. Ряд стратегий проектирования сетей

В этой группе методов изображения с низким разрешением сначала интерполируются для получения «грубых» изображений с большим разрешением. Теперь CNN используются для обучения end-to-end отображения от интерполированных изображений с низким качеством до изображений с высоким качеством. Идея этой группы методов состоит в

том, что может быть проще сначала увеличить разрешение изображений, используя традиционные методы (такие как билинейная интерполяция), а затем сделать результирующий результат, чем изучить прямое отображение из низкоразмерного пространства в многомерное пространство (рис. 4).

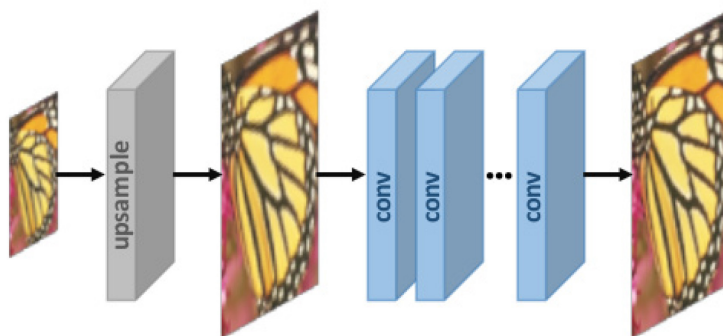


Рис.4. Обыкновенная Pre-Upsampling сеть

Пост-увеличение разрешения (Post-Upsampling)

В этом случае на CNN подаются оригинальные изображения с низким разрешением. Повышение разрешения выполняется на последнем обучаемом слое (рис. 5).

Преимущество этого метода состоит в том, что извлечение признаков выполняется в пространстве меньшего размера (перед повышением разрешения), и, следовательно, вычислительная сложность уменьшается. Кроме того,

используя обучаемый слой upsampling, модель можно обучать end-to-end.

Прогрессивное увеличение разрешения (Progressive Upsampling)

В группе методов с пост-увеличением разрешения, хотя вычислительная сложность и была снижена, но использовалась только одна свертка с повышением размерности.

Это усложняет процесс обучения при больших коэффициентах масштабирования.

Для устранения этого недостатка, в таких работах, как сеть Laplacian Pyramid SR (Lap SRN) и Progressive SR (Pro SR), была принята концепция прогрессивного увеличения разрешения. Модели в этом случае используют кас-

кад CNN для постепенного восстановления изображений с высоким разрешением при меньших коэффициентах масштабирования на каждом этапе (рис. 6).

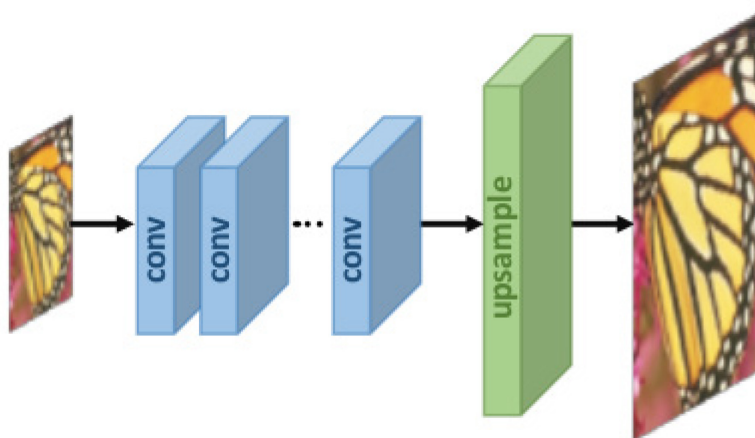


Рис.5. Пост-увеличение разрешения (Post-Upsampling)

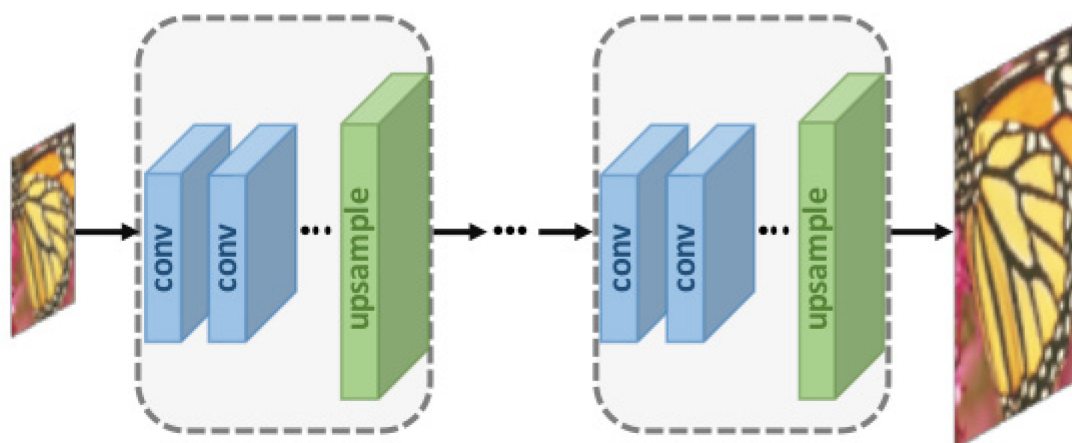


Рис. 6. Прогрессивное увеличение разрешения (Progressive Upsampling)

Раскладывая сложную задачу на более простые, трудность в обучении значительно снижается, и можно добиться лучшей производительности. Кроме того, стратегии обучения, такие как обучение по учебному плану, могут быть интегрированы для дальнейшего снижения сложности обучения и повышения итоговой успеваемости.

Итеративное понижение и повышение разрешения

Другой популярной моделью архитектуры является структура песочных часов (или U-Net). В некоторых вариантах, таких как сеть Stacked Hourglass, используется несколько последовательных структур песочных часов, ко-

торые эффективно чередуются между процессами повышения и понижения размерности (рис. 7).

Модели в этой структуре могут лучше определить глубокие отношения между парами изображений LR-HR и, таким образом, обеспечить более качественные результаты реконструкции.

Не редкой в GAN используются ошибки текстуры. Чтобы позволить сгенерированному изображению иметь тот же стиль (текстуру, цвет, контраст и т.д.), что и у настоящего изображения, используется вычисление ошибки текстуры (или ошибка реконструкции стиля). Текстура изображения, определяется как корреляция между различными характерными каналами. Каналы объектов обычно получают

из карт признаков, извлеченных с использованием предварительно обученной сети классификации изображений (Φ).

$$G_{ij}^{(l)}(I) = \text{vec}(\phi_i^{(l)}(I)) \cdot \text{vec}(\phi_j^{(l)}(I)).$$

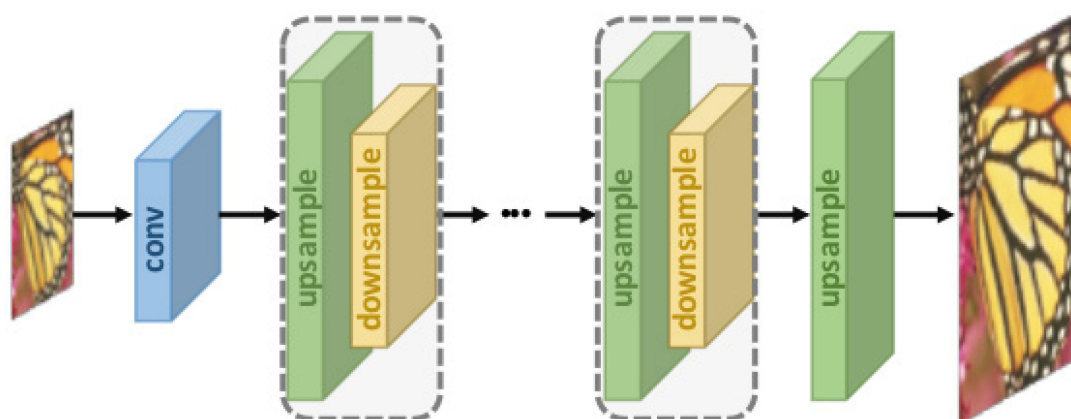


Рис. 7. Структура песочных часов или U-Net

Корреляция между картами признаков представлена **матрицей Грама** (G), которая является внутренним произведением между векторизованными картами объектов i и j на

слое l . Как только матрица Грамма рассчитана для обоих изображений, вычисляется ошибка текстуры:

$$\mathcal{L}_{\text{texture}}(\hat{I}, I; \phi, l) = \frac{1}{c_l^2} \sqrt{\sum_{i,j} (G_{i,j}^{(l)}(\hat{I}) - G_{i,j}^{(l)}(I))^2}.$$

Используя эту ошибку, модель мотивируется создавать реалистичные текстуры и визуально более удовлетворительные результаты [7].

Ошибка суммарного отклонения (TV) используется для подавления шума в сгенериро-

ванных изображениях. Он берет сумму абсолютных разностей между соседними пикселями и измеряет, сколько шума на изображении. Для сгенерированного изображения ошибки TV рассчитываются, как показано ниже:

$$\mathcal{L}_{\text{TV}}(\hat{I}) = \frac{1}{hwc} \sum_{i,j,k} \sqrt{(\hat{I}_{i,j+1,k} - \hat{I}_{i,j,k})^2 + (\hat{I}_{i+1,j,k} - \hat{I}_{i,j,k})^2}.$$

Здесь i, j, k перебирает высоту, ширину и каналы соответственно.

Один большой вопрос заключается в том, как можно количественно оценить производительность модели. Ряд методов оценки качества изображения (IQA) (или метрики) используются для одного и того же. Эти метрики можно в целом классифицировать на две категории - субъективные метрики и объективные метрики.

Субъективные метрики основаны на оценке восприятия человеком-наблюдателем, тогда

как объективные метрики основаны на вычислительных моделях, которые пытаются оценить качество изображения. Субъективные метрики часто являются более «точными для восприятия», однако некоторые из этих метрик неудобны, трудоемки или дороги для вычисления. Другая проблема заключается в том, что эти две категории метрик могут не соответствовать друг другу. Следовательно, исследователи часто анализируют результаты, используя метрики из обеих категорий.

PSNR

Пиковое отношение сигнал/шум (PSNR) — это обычно используемая объективная метрика для измерения качества восстановления преобразования с потерями. PSNR обратно пропорционально логарифму средней квадратичной ошибки (MSE) между настоящим и сгенерированным изображением.

$$MSE = \frac{1}{N} \sum_{i=1}^N (I(i) - \hat{I}(i))^2,$$

$$PSNR = 10 \cdot \log_{10} \left(\frac{L^2}{MSE} \right).$$

В приведенной выше формуле L — максимально возможное значение пикселя (для 8-

битных изображений RGB — 255). Неудивительно, что, поскольку PSNR заботится только о разнице между значениями пикселей, он не так хорошо отражает воспринимаемое качество.

SSIM

Структурное сходство (SSIM) — это субъективная метрика, используемая для измерения структурного сходства между изображениями на основе трех относительно независимых сравнений, а именно: яркости, контрастности и структуры. Абстрактно, формула SSIM может быть показана как взвешенное произведение яркости, контраста и структуры, вычисленных независимо.

$$SSIM(I, \hat{I}) = [C_l(I, \hat{I})]^\alpha [C_c(I, \hat{I})]^\beta [C_s(I, \hat{I})]^\gamma,$$

В приведенной выше формуле альфа, бета и гамма являются весами функций сравнения яркости, контраста и структуры соответствен-

но. Обычно используемое представление формулы SSIM показано ниже:

$$SSIM(I, \hat{I}) = \frac{(2\mu_I \mu_{\hat{I}} + C_1)(\sigma_{I\hat{I}} + C_2)}{(\mu_I^2 + \mu_{\hat{I}}^2 + C_1)(\sigma_I^2 + \sigma_{\hat{I}}^2 + C_2)}.$$

В приведенной выше формуле $\mu(I)$ представляет среднее значение конкретного изображения, $\sigma(I)$ представляет стандартное отклонение конкретного изображения, $\sigma(I, I')$ представляет ковариацию между двумя изображениями и C_1, C_2 задают константы, чтобы избежать нестабильности. Структурное сходство может быть использовано при решении инженерных, эргономических и дизайнерских творческих задач [8].

Заключение и выводы

Из-за возможного неравномерного распределения статистических характеристик или искажений изображения локальная оценка качества изображений надежнее, чем его оценка в глобальном масштабе. Среднее значение SSIM (MSSIM), которое разбивает изображение на несколько окон и усредняет значение

SSIM, полученное в каждом окне, является одним из таких методов оценки качества на локальном уровне.

В любом случае, SSIM оценивает качество реконструкции с точки зрения зрительной системы человека, эта метрика лучше соответствует требованиям оценки восприятия.

В статье были рассмотрены некоторые вводные материалы и процедуры обучения моделей глубокого обучения для выполнения задачи супер-разрешения. Были рассмотрены методики, которые могут дать лучшую производительность супер-разрешения без учителя, лучших методов нормализации и лучших репрезентативных метрик, которые могут значительно расширить область понимания генеративно-сопоставительных сетей. Перспективным направлением исследования является анализ.

СПИСОК ЛИТЕРАТУРЫ

1. Аггарвал, Ч. Нейронные сети и глубокое обучение: учебный курс: Перевод с английского [Текст] / Ч. Аггарвал. – СПб.: ООО “Диалектика”, 2020. – 752 с. – ISBN 978-5-907203-01-3.
2. Жадаев, Д.С. Особенности нейросетевого анализа уровня подготовки студентов в процессе адаптивного тестирования их профессиональных компетенций [Текст] / Д.С. Жадаев, А.А. Кузьменко, В.В. Спасенников // Вестник Брянского государственного технического университета – 2019. – №2(75). – С. 90-98.
3. Николенько, С. Глубокое обучение. Погружение в мир нейронных сетей [Текст] / С. Николенько, А. Кадурин, Е. Архангельская. – СПб.: Издательский дом «Питер», 2018. – 482 с.
4. Патент RU 2309457 Модель нейронной сети МПК G06N3/06 / К.Н. Шевченко, Н.В. Шевченко, Б.В. Шульгин Приоритет 06.05.2006, Опубликовано 27.10.2007, БИ. №30.
5. Гудфеллоу, И. Дж. Генеративные состязательные сети, в: Advances в системах обработки нейронной информации [Текст] / И. Дж. Гудфеллоу, Дж. Пуже-Абади, М. Мирза, Б. Сюй, Д. Вард-Фарли, С. Озер, А.С. Курвиль, Ю. Бенжю // 27-я Ежегодная конференция по нейронным системам. – 2014. – С. 2672-2680.
6. Бокс, Дж. Анализ временных рядов: прогнозирование и контроль [Текст] / Дж. Бокс, Д.М. Дженкинс // Journal of Time. – №31. – 1976. – С. 238-242.
7. Дин, Х. Глубокое обучение для прогнозирования акций на основе событий, в: Материалы Двадцать четвертой Международной совместной конференции по искусственному интеллекту [Текст] / Х. Дин, Ю.Чжан, Т. Лю, Дж. Дуань // IJCAI, 2015. – С. 2327-2333.
8. Синюй, Чж, Прогноз фондового рынка. Высоочастотные данные с использованием генеративных состязательных сетей [Текст] / Чж. Синюй, П. Чжисун, Х. Гуо, Т. Сыци, Ч. Чжао // Математические проблемы. Инжиниринг. – 2018. – С. 7-10.
9. Разер, А.М. Рекуррентная нейронная сеть и гибридная модель для прогнозирования доходности акций [Текст] / А.М. Разер, А. Аггарвал, В.Н. Састри // Экспертная система. Appl., - №42. – 2015. - С. 3234-3241.
10. Цантэкидис, А. Прогнозирование цен на акции из книги лимитных заявок с помощью сверточных нейронных сетей [Текст] / А. Цантэкидис, Н. Пассалис, А. Тефас, Дж. Канниайнен, М. Габбудж, А. Иосифидис // 19-е. Конференция IEEE по бизнес-информатике, CBI 2017. – С. 7–12.
11. Zeiler, M. Visualising and understanding convolution networks. European Conference on Computer Vision [Электронный ресурс] / M. Zeiler, R. Fergus. – URL: <https://arxiv.org/pdf/1311.2901.pdf> (дата обращения 29.09.2020).
12. Zhang, H. Character-level convolutional networks for text classification [Текст] / H. Zhang, J. Zhao, Y. Le Cun // NIPS Conference. – 2015. - С. 649-657.

REFERENCES

1. Aggarwal, Ch. Neural networks and deep learning: course English [Text] / Ch. Aggarwal. – SPb: LLC “Dialectic”, 2020. – 752p. – ISBN 978-5-907203-01-3.
2. Zhadaev, D. S. Features neural network analysis of level of preparation of students in the process of adaptive testing of their professional competence [Text] / D. S. Zhadaev, A. A. Kuzmenko, V. V. Spasennov // Bulletin of the Bryansk state technical University. – 2019. - №2 (75). – P. 90-98.
3. Nikolenko, S. Deep learning. Immersion in the world of neural networks [Text] / S. Nikolenko, A. Kadurin, E. Arkhangelskaya. - SPb.: Publishing house "Peter", 2018. – 482 p.
4. Patent RU 2309457 Model of the neural network IPCG06N3/06 / K. N. Shevchenko, N. V. Shevchenko, B. V. Shulgin Priority 06.05.2006, published 27.10.2007, BI. no. 30.
5. Goodfellow, I. J. Generative competitive networks, in: Advances in neural information processing systems [Text] / I. J. Goodfellow, J. Pouge-Abadi, M. Mirza, B. Xu, D. Ward-Farley, S. Ozer, A. S. Courville, Y. Benjio // 27th annual conference on neural systems. - 2014. – P. 2672-2680.
6. Box, J. Analysis of time series: forecasting and control [Text] / J. Box, D. M. Jenkins // Journal of Time. - No. 31. – 1976. – С. 238-242.
7. Ding, X. Deep learning for predicting stocks based on events, in: Proceedings of the Twenty-fourth International joint conference on artificial intelligence [Text] / H. Ding, Yu. Zhang, T. Liu, J. Duan // IJCAI, 2015. – P. 2327-2333.
8. Sinyu, Zhi, Stock market forecast. High-frequency data using generative adversarial networks [Text] / Zh. Sinyu, P. Zhixun, H. Guyu, T. Siqi, Ch. Zhao // Mathematical problem. Engineering. – 2018. – P. 7-10.
9. Razer, A.M. Recurrent neural network and hybrid model for predicting stock returns [Text] / A.M. Razer, A. Agarwal, V. N. Sastri // Expert system. Appl., - №42. – 2015. - P. 3234-3241
10. Tsantekidis, A. Forecasting stock prices from the book of limit orders using convolutional neural networks [Text] / A. Tsantekidis, N. Passalis, A. Tefas, J. Kannianen, M. Gabbudzh, A. Iosifidis // 19-e. IEEE conference on business Informatics, CBI 2017. – P. 7-12.
11. Zeiler, M. Visualising and understanding convolution networks. European Conference on Computer Vision [Electronic resource] / M. Zeiler, R. Fergus. – URL: <https://arxiv.org/pdf/1311.2901.pdf> (accessed 29.09.2020).
12. Zhang, H. Character-level convolutional networks for text classification [Text] / H. Zhang, J. Zhao, Y. Le Cun // NIPS Conference. – 2015. - C. 649-657.

Ссылка для цитирования: Малахов Ю.А. Анализ и применение генеративно-сопоставительных сетей для получения изображений высокого качества / Ю.А. Малахов, А.А. Андросов, А.В. Аверченков // Эргодизайн. – 2020 - №4 (10). – С. 167-176. - DOI: 10.30987/2658-4026-2020-4-167-176.

Сведения об авторах:

Малахов Юрий Антонович

доцент кафедры «Компьютерные технологии и системы» Брянского государственного технического университета

E-mail: yumal55@yandex.ru

ORCID 0000-0001-8474-0927

Андросов Александр Андреевич

аспирант Брянского государственного технического университета

E-mail: shkafiara@bk.ru

Аверченков Андрей Владимирович

профессор кафедры «Компьютерные технологии и системы» Брянского государственного технического университета

E-mail: mahar@mail.ru

Abstracts:

Y.A. Malakhov

Associate Professor, Department of Computer Technologies and Systems, Bryansk State Technical University

E-mail: yumal55@yandex.ru

ORCID 0000-0001-8474-0927

A. A. Androsov

master of Bryansk State Technical University

E-mail: shkafiara@bk.ru

A. V. Averchenkov

Professor, Department of Computer Technology and Systems, Bryansk State Technical University

E-mail: mahar@mail.ru

Статья поступила в редколлегию 29.09.2020 г.

Рецензент:

д.т.н., профессор Санкт-Петербургского государственного технического университета

(«ЛЭТИ им. В.И. Ульянова-Ленина»)

член редакционного совета журнала "Эргодизайн"

Падерно П.И.

Статья принята к публикации 10.10.2020 г.