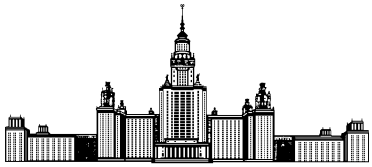

Московский Государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

Гитман Игорь Александрович

Применение конкурирующих сетей для задач машинного обучения

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:
к.ф-м.н.,
Д.П. Ветров

Москва, 2016

Аннотация

В рамках данной работы было проведено исследование применимости модели конкурирующих сетей для решения задачи устранения размытости изображений. В ходе исследования было разработано три различных подхода для решения данной задачи, проведен анализ недостатков каждого из подходов и экспериментально установлены границы их применимости. Также на основе проведенного исследования был разработан метод обучения нейронных сетей для решения задачи устранения размытости изображений за счет минимизации функции ошибки, порождаемой глубинной сверточной нейронной сетью, предобученной для решения задачи классификации изображений. В конце работы приводится экспериментальное сравнение предложенного подхода и современных методов для решения данной задачи, где демонстрируется, что предложенный в работе метод способен восстанавливать визуально более качественные изображения.

Оглавление

1 Введение

Задача восстановления искаженных изображений является одной из основных задач в области обработки изображений. На практике, из-за движения камеры во время снимка, либо неправильной настройки фокуса, часто возникают искажения вида частичного или полного размытия изображений. Задача автоматического устранения подобного рода артефактов называется задачей устранения размытости изображений.

В общем случае задача устранения размытости изображений некорректно поставлена по Адамару (имеет множество решений). Поэтому большинство современных методов регуляризируют исходную задачу за счет введения априорной информации о том, как выглядят фотореалистичные изображения. Это может быть введение априорного распределения на градиенты изображений [1], использование статистической информации о цветовых переходах [2], введение ограничений на геометрию границ [3] и др. Также в последние годы, в связи с большим успехом глубинных сверточных нейронных сетей [4] в различных задачах компьютерного зрения ([5], [6], [7]), были разработаны эффективные методы их применения и для решения задачи устранения размытости изображений ([8], [9]).

В рамках данной работы было проведено исследование возможностей объединения двух озвученных подходов. Был предложен способ введения априорной информации о фотореалистичных изображениях в методы, основанные на глубинных нейронных сетях, за счет использования модели конкурирующих сетей [10]. Конкурирующие сети на сегодняшний день являются одной из лучших моделей для генерации фотореалистичных изображений. В рамках данной модели генерация производится с помощью «генеративной нейронной сети», которая преобразует случайный шум в изображения. Функция ошибки для обучения данной сети порождается «дискриминативной нейронной сетью», которая обучается детектировать структурные различия между генерируемыми и настоящими изображениями и таким образом «направляет» обучение генеративной сети. Если в качестве генеративной сети использовать нейронную сеть, способную решать задачу устранения размытости изображений (то есть, принимающую на вход искаженные изображения и возвращающую восстановленные), то можно говорить о том, что использование конкурирующего обучения ведет к регуляризации исходной задачи. Действительно, стандартная процедура обучения заключается в минимизации L_2 нормы разницы между исходными и восстановленными изображениями, что поощряет получение частично размытых результатов в сложных областях. Более подробно данная проблема описана в [11], [12]. В случае конкурирующего обучения, дискриминативная сеть сможет автоматически детектировать излишнюю размытость восстановленных изображений, что в итоге может сделать их более резкими.

В рамках данной работы было экспериментально показано, что применение конкурирующих сетей для решения задачи устранения размытости изображений сопряжено с определенными трудностями. При замене стандартной функции ошибки на порождаемую дискриминативной сетью, восстановленные изображения действительно становятся более резкими, но перестают быть похожими на исходные. Подобный эффект можно объяснить тем, что при использовании стандартного конкурирующего обучения генеративная сеть не получает никакой информации об исходных изображениях и лишь переводит размытые изображения в некоторые фотореалистичные. Гипотеза о том, что наиболее простым фотореалистичным изображением, которое можно получить из размытого, будет исходное, оказалась неверной.

Для устранения озвученных недостатков нами был разработано два различных подхода, объединяющих конкурирующее и стандартное обучение нейронных сетей. Стандартное обучение нейронных сетей заключается в минимизации L_2 нормы разницы между исходными и восстановленными изображениями. Конкурирующее обучение заключается в

минимизации логарифма отклика дискриминативной сети. Первый подход заключался в использовании функции ошибки, состоящей из взвешенной суммы данных двух слагаемых. При таком подходе восстановленные изображения действительно получаются похожими на исходные, а также в целом более резкими, чем без использования конкурирующего обучения. Но резкость достигается за счет добавления на изображения небольших штрихов, вписывающихся в маленькие детали, которые в большинстве случаев выглядят как случайный шум.

Второй из разработанных подходов основывается на наблюдении, что на последних слоях дискриминативной сети формируется скрытое представление поданных ей на вход изображений. Нами была выдвинута гипотеза, что осмысленно будет явно минимизировать не только по-пиксельные различия между исходными и восстановленными изображениями, но и различия между скрытыми представлениями, формируемыми дискриминативной сетью. Действительно, так как дискриминативная сеть может определять, что восстановленные изображения в целом более размытые, чем исходные, то эта информация должна содержаться в их скрытых представлениях. Если мы потребуем, чтобы скрытые представления стали более похожими, то и полученные изображения станут более похожими с точки зрения дискриминативной сети, что может привести к увеличению резкости. Таким образом, в данном подходе также предлагается минимизировать взвешенную сумму, но вместо стандартного для конкурирующих сетей логарифма отклика дискриминативной сети использовалась L_2 норма разницы скрытых представлений исходных и восстановленных картинок, формируемых дискриминативной сетью. Заметим, что при таком подходе в конкурирующее обучения явно вносится информация о сопоставлении резких и размытых изображений, что позволяет модели настраиваться не только на различия двух распределений в целом, но и на различия между конкретными изображениями. Однако на практике подобный подход оказался не слишком эффективным. Экспериментально было продемонстрировано, что данный метод выдает более резкие изображения, чем первый из озвученных подходов, но в целом подвержен тем же недостаткам.

Анализируя проведенное исследование можно сделать вывод о целесообразности добавления в предложенные модели информации о конкретном виде объектов, расположенных на изображениях. Действительно, восстановленные изображения, содержащие равномерный фон (например, небо) могут быть сильно более размытыми, чем изображения, содержащие резкие текстурные элементы (например, шерсть животных). Для того, чтобы внести в модель подобную информацию нами была предложена модификация последнего из озвученных подходов, заключающаяся в замене дискриминативной сети на глубинную сверточную нейронную сеть VGG-16 [13], предобученную для классификации изображений. В качестве функции ошибки генеративной сети использовалась L_2 норма разницы скрытых представлений исходных и восстановленных изображений, формируемых данной сетью. Также, мы отказались от конкурирующего обучения, так как существующие методы обучения конкурирующих сетей не позволяют работать с настолько глубокими нейронными сетями (разработка подобных алгоритмов может быть одним из путей развития данной работы). Тем не менее экспериментально было показано, что предложенный метод эффективно работает и резкость на изображениях повышается не равномерно, а в зависимости от расположенных на них объектах. Также было проведено экспериментальное сравнение с современными алгоритмами для решения задачи устранения размытости и показано, что предложенных метод во многих случаях восстанавливает более резкие и фотореалистичные изображения.

2 Задача устранения размытости изображений

2.1 Общая постановка задачи

В наиболее общем виде модель размытия изображений формулируется следующим образом:

$$y = \phi(x * k + n)$$

Здесь $x, y, n \in \mathcal{R}^{p \times q}, k \in \mathcal{R}^{p' \times q'}, \phi : \mathcal{R}^{p \times q} \to \mathcal{R}^{p \times q}$. x — это исходное изображение размера $p \times q$, $*$ — оператор свертки, k — ядро свертки размера $p' \times q'$, n — аддитивный случайный шум, ϕ — некоторое нелинейное преобразование (например, частичное затухание яркости или JPEG компрессия), y — искаженное (размытое) изображение. Задача устранения размытости: по данным y и k восстановить исходное изображение x[1]. Без ограничения общности можно считать, что все операции проводятся с одноканальными изображениями (в случае трехканальных изображений, каждый канал может обрабатываться отдельно).

Заметим несколько особенностей такой постановки задачи. Если мы положим $\phi(x) = x$, $n = 0$ (отсутствие шума и нелинейных преобразований), то задача может быть решена аналитически. Действительно, используя теорему о свертке мы можем переписать уравнение на x в следующем виде:

$$y = x * k \Leftrightarrow \mathcal{F}(y) = \mathcal{F}(x) \cdot \mathcal{F}(k) \Rightarrow x = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(y)}{\mathcal{F}(k)}\right) = \mathcal{F}^{-1}\left(\frac{1}{\mathcal{F}(k)}\right) * y$$

где $\mathcal{F}(\cdot)$ — это дискретное преобразование Фурье, $\mathcal{F}^{-1}(\cdot)$ — обратное дискретное преобразование Фурье, а операции деления и умножения выполняются поэлементно. В случае, когда в модели присутствует шум, задача становится некорректно поставленной по Адамару, так как решений x, удовлетворяющих предложенной модели становится бесконечно много.

2.2 Обзор существующих методов решения

Методы для решения данной задачи различаются как рассматриваемыми моделями размытия, так и подходом к нахождению оптимального восстановленного изображения для выбранной модели размытия. Условно разделим все существующие методы на 4 группы:

1. Классические методы;
2. Методы, использующие априорную информацию;
3. Методы, явно моделирующие нелинейность;
4. Нейросетевые методы.

2.3 Классические методы

Данная группа методов предполагает отсутствие нелинейности в модели размытия, а также конкретный вид шума. Восстановленное изображение находится по методу максимального правдоподобия, что в случае Гауссовского распределения шума эквивалентно

[1] данная задача в англоязычной литературе называется non-blind image deconvolution в противовес blind image deconvolution, когда ядро свертки k считается неизвестным.

Рис. 2.1: Пример работы простых алгоритмов в ситуации нарушения предположений их моделей (яркость на картинке частично затухает). Слева-направо: исходное изображение, размытое изображение, восстановленное алгоритмом Люси-Ричардсона, восстановленное алгоритмом Вейнера. Видно, что алгоритмы либо выдают слишком размытое изображение, либо добавляют артефакты-кольца.

решению следующей оптимизационной задачи:

$$y = x * k + n,\ n \sim \mathcal{N}(0, I\sigma^2)$$
$$\hat{x} = \arg\min_x \left[\|x * k - y\|^2 \right]$$

здесь и далее $\|\cdot\|$ обозначает Фробениусову норму матрицы (L_2 норму для векторов) , σ^2 — некоторый фиксированный параметр, задающий степень зашумленности изображений. В этом случае (а также в более общих предположениях о независимости шума от исходной картинки) можно получить аналитическую формулу для оптимального восстановленного изображения:

$$x = \mathcal{F}^{-1}\left(\frac{1}{\mathcal{F}(k)} \left[\frac{|\mathcal{F}(k)|^2}{|\mathcal{F}(k)|^2 + \frac{1}{SNR}} \right] \right) * y$$

где SNR — это отношение сигнал/шум, которое в практических задачах неизвестно и необходимо как-то оценивать. Такой способ решения задачи устранения размытости называется деконволюцией Вейнера [14]. Если же мы предположим, что шум имеет Пуассоновское распределение, то можно получить итерационный алгоритм восстановления картинки, получивший название алгоритма Люси-Ричардсона ([15], [16])[2]. Полученные алгоритмы работают очень быстро, но крайне чувствительны даже к небольшим отклонениям от выбранной модели. Во многих случаях на восстановленных изображениях наблюдаются заметные артефакты или изображения получаются недостаточно резкими (рис. 2.1).

2.4 Методы, использующие априорную информацию

Данная группа методов использует априорную информацию о том, как выглядят настоящие изображения, чтобы сделать алгоритмы более устойчивыми к отклонениям от выбранной модели размытия, а также повысить резкость получаемых результатов. Так как исходная задача некорректно поставлена, ее решениями могут являться как резкие настоящие изображения, так и частично размытые и зашумленные. Добавление подходящих регуляризаторов приводит к тому, что восстановленные изображения становятся более фотореалистичными.

Методы из данной группы явно не моделируют нелинейность, а также обычно работают в предположении, что шум распределен по Гауссовскому закону, но за счет добавления априорной информации становятся более стабильными и выдают более качественные результаты на реальных изображениях. В качестве восстановленной картинки используется

[2] эти два алгоритма по праву можно считать классическими алгоритмами для устранения размытости изображений, хотя они, конечно, уступают в качестве современным методам.

мода апостериорного распределения, что эквивалентно решению следующей оптимизационной задачи:

$$y = x * k + n,\ n \sim \mathcal{N}(0, I\sigma^2)$$
$$\hat{x} = \arg\min_x \left[\|x * k - y\|^2 + R(x) \right]$$

где $R(x)$ — некоторый регуляризатор, который задает наши априорные знания о фотореалистичных изображениях. Это может быть распределение на градиенты фотореалистичных изображений [1], статистическая информация о цветовых переходах [2], ограничения на геометрию границ [3], нормированные разреживающие регуляризаторы [12] и др. Полученные оптимизационные задачи в каждом отдельном случае решаются различными методами, зависящими от вида регуляризатора.

Стоит отметить, что в различных случаях использование одного и того же априорного распределения может способствовать как получению визуально реалистичных, так и совершенно нереалистичных изображений (рис. 2.2). Поэтому, выбор подходящего априорного распределения, на сегодняшний день является одним из ключевых факторов для успешного решения задачи устранения размытости изображений.

2.5 Методы, явно моделирующие нелинейность

Методы из данной группы также используют априорные распределения на фотореалистичность изображений, но кроме этого еще и явно включают в модель различные виды нелинейных преобразований, характерные для размытия изображений, полученных с помощью фотосъемки. В основном, методы из данной группы моделируют размытие следующим образом:

$$y = \phi(x * k + n)$$
$$\hat{x} = \arg\min_{\tilde{x} \subseteq x} \left[\|x * k - y\|^2 + R(x) \right]$$

приведенную выше формулу следует понимать не в строгом смысле, а лишь как иллюстрацию идеи, используемой в данных методах. То есть, данные методы работают в предположении, что лишь некоторое подмножество пикселей в исходном изображении не удовлетворяет стандартной линейной модели с Гауссовским шумом. Эти пиксели обрабатываются отдельным образом, а значения остальных ($\tilde{x}$) восстанавливаются с помощью нахождения максимума апостериорной вероятности, как и для предыдущей группы методов. Подобное предположение будет выполняться, например, при неравномерном затухании яркости, которое вызвано тем, что современные камеры не могут работать в условиях освещения произвольной интенсивности и засекают только определенный диапазон яркостей. Если на сенсор камеры попадает больше фотонов света, чем она может засечь, то вместо реального значения интенсивности будет записано максимально возможное (формально, к изображению применяется нелинейная функция $c(x) = \max(x, L)$, где L — максимальная яркость, которую камера может измерить). Ясно, что применение такого преобразования затронет только небольшое подмножество пикселей исходного изображения, находящихся рядом с очень яркими источниками света, поэтому предположения методов часто являются обоснованными.

Для вывода в подобных моделях обычно используются схемы, в которых в начале некоторым образом оценивается множество пикселей, являющихся выбросами (не удовлетворяющих линейной модели), а затем в качестве восстановленного изображения используется максимум апостериорной вероятности, с учетом равномерного распределения на выбросы. Хармелинг и др. [17] предлагают относительно простую схему, в которой пиксели, значения яркости которых выше некоторого порога не учитываются при оценке восстановленного изображения. Чо и др. [18] используют EM-алгоритм, в котором на E-шаге оценивается какие пиксели на изображении являются выбросами, а на M-шаге

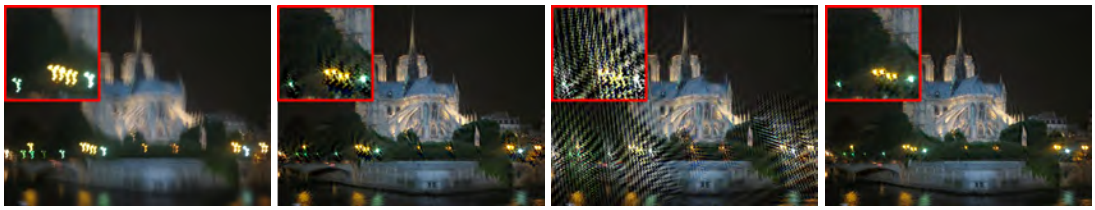

Рис. 2.2: Пример, когда алгоритмы явно учитывающие нелинейность в модели работают лучше методов из второй и первой групп. Слева-направо: размытое изображение с затуханием яркости, восстановленное алгоритмом Люси-Ричардсона, восстановленное методом [12], восстановленное методом [19], учитывающим нелинейности данного типа. Также, на этом примере видно, что алгоритмы, использующие априорные распределения могут в некоторых ситуациях восстанавливать визуально нереалистичные изображения.

проводится устранение размытости, не учитывая эти пиксели. Вайт и др. [19] разработали итерационный алгоритм по типу алгоритма Люси-Ричардсона, способный явно учитывать неравномерное затухания яркости при выводе в предложенной модели (то есть, в этом алгоритме нет необходимости оценивать какие пиксели являются выбросами), используя гладкую аппроксимация функции $c(x)$.

Во многих случаях методы из данной группы демонстрируют лучшее качество, чем методы, которые явно не моделируют нелинейности (рис. 2.2).

2.6 Нейросетевые методы

В последние годы наилучшие результаты во многих задачах компьютерного зрения были достигнуты с помощью использования глубинных сверточных нейронных сетей ([5], [6], [7]). Нейросетевой подход к задаче устранения размытости изображений может быть сформулирован следующим образом:

$$y = \phi(x * k + n)$$
$$\hat{x} = f(y; \hat{\theta})$$
$$\hat{\theta} = \arg\min_{\theta} \left(\frac{1}{n} \sum_{i=1}^{n} \left\| x_i - f(y_i; \theta) \right\|^2 \right)$$

где $f(y;\theta)$ — это нейронная сеть, которая переводит размытую картинку в резкую, $\{x_i, y_i\}_{i=1}^{n}$ — обучающая выборка, состоящая из размытых и соответствующих им резких изображений. Данный подход предпочтителен тем, что мы можем генерировать почти неограниченную выборку размытых изображений, причем как шум, так и нелинейная функция, в отличии от предыдущих методов, могут быть практически произвольными (единственное требование, это возможность применения преобразований к резким изображениям для генерации выборки). Подобные подходы активно применяются для решения задачи удаления шума с изображений [20] или увеличения разрешения [7]. Но не смотря на внешнее сходство с озвученными задачами, стандартные нейросетевые методы оказываются неприменимы к задаче устранения размытости изображений. Стандартные нейросетевые архитектуры оказываются не способны выучить необходимые преобразования и восстановленные изображения получаются слишком размытыми [8]. Также, на текущий момент не существует эффективных способов параметризации нейросетевых моделей ядром свертки, что ведет к необходимости обучения новой сети для каждого отдельного типа ядра и сильно сужает границы применимости подобных методов.

Тем не менее, исследования последних лет показали, что нейросетевые подходы имеют большой потенциал: Шулер и др. [9] предложили использовать нейронные сети в качестве пост-обработки изображений для устранения артефактов, производимых другими

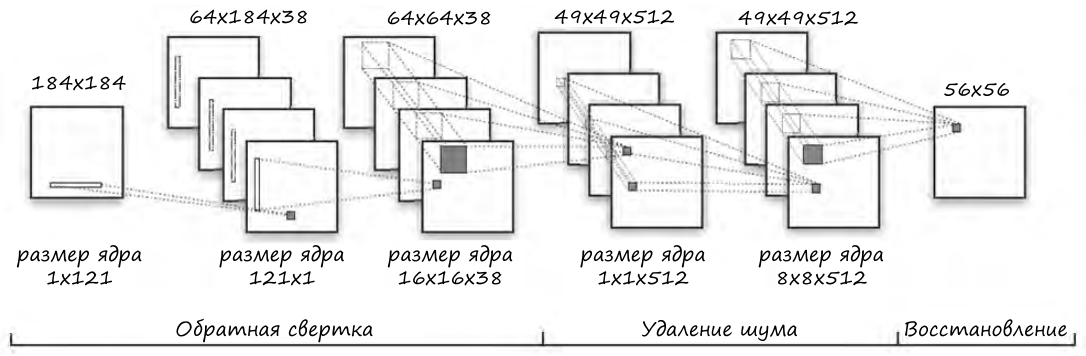

Рис. 2.3: Архитектура сверточной нейронной сети для решения задачи устранения размытости изображений, предложенная в [8]. Первая часть сети состоит из вертикальных и горизонтальных 1D фильтров, которые моделируют обратное ядро свертки. Вторая часть сети устраняет шум с изображения.

алгоритмами, а Ху и др. [8] разработали эффективную нейросетевую архитектуру, способную восстанавливать резкие изображения лучше, чем многие другие методы для решения данной задачи. Рассмотрим подробнее предложенную авторами архитектуру, так как подобная архитектура существенно использовалась в данной работе.

Анализируя точное решение задачи устранения размытости, а также алгоритм Вейнера, авторы статьи пришли к выводу, что исходное резкое изображение может быть хорошо приближено сверткой искаженного изображения с некоторым ядром (возможно, размера порядка размера исходной картинки):

$$x = \mathcal{F}^{-1}\left(\frac{1}{\mathcal{F}(k)}\left[\frac{|\mathcal{F}(k)|^2}{|\mathcal{F}(k)|^2 + \frac{1}{SNR}}\right]\right) * y = \hat{k} * y$$

$\hat{k}$ будет называть обратным ядром свертки. Используя сингулярное разложение $\hat{k} = USV^T$, а также линейность операции свертки, можно получить следующее выражение для восстановленного изображения:

$$\hat{k} * y = \sum_j s_j u_j * (v_j^T * y)$$

где u_j, v_j — j-ые столбцы матриц U и V соответственно, s_j — j-ое сингулярное число. Данное выражение показывает, что 2D свертка может быть представлена в виде взвешенной суммы независящих друг от друга 1D фильтров. На практике, фильтры, соответствующие сингулярным числам $s_j < 0.01$ можно игнорировать. Также, экспериментально было показано, что при размере обратного ядра $\approx 100 \times 100$ пикселей (соответствующих высоким частотам в частотном пространстве) визуально результаты получаются неотличимыми от использования полного обратного ядра свертки. Подобные соображения позволили авторам статьи спроектировать эффективную архитектуру сверточной нейронной сети для решения задачи устранения размытости изображений (рис. 2.3).

Таким образом, наилучшие результаты при решении задачи устранения размытости изображений можно получить либо за счет использования хорошо подобранного априорного распределения и аккуратного учета множества факторов (различных нелинейных преобразований, влияния выбросов и др.), либо за счет использования больших обучающих выборок и нейросетевых подходов, которые способны автоматически учитывать все необходимые факторы. Однако, даже при использовании специальных архитектур, нейросетевые методы генерируют частично размытые изображения. Дело в том, что минимизация L_2 нормы поощряет размытие в сложных областях, что подтверждается экспериментально и было теоретически обосновано в [11], [12]. В рамках данной работы проводится

исследование возможностей улучшения нейросетевых подходов за счет добавления в них априорной информации о том, как выглядят фотореалистичные изображения. Для этого была использована недавно предложенная модель конкурирующих сетей [10], способная автоматически выучивать сложные распределения с единственным предположением, что мы можем генерировать выборки достаточно большого объема.

3 Модель конкурирующих сетей

3.1 Сверточные нейронные сети

Сверточные нейронные сети — это нейронные сети, специально спроектированные для того, чтобы учитывать структурные особенности объектов, которые подаются им на вход. Наиболее эффективно подобные сети работают с объектами-изображениями (идея подобных нейронных сетей возникла при анализе зрительной коры животных). Стандартная сверточная нейронная сеть состоит из цепочки сверточных слоев, которые используются для автоматического выделения признаков, и нескольких полносвязных слоев, которые используются для решения поставленной задачи, например, классификации изображений (рис. 3.1).

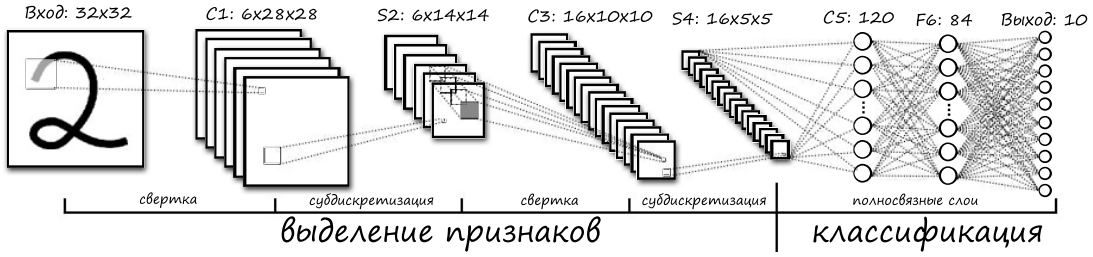

Рис. 3.1: Архитектура сверточной сети LeNet-5 для классификации цифр.

Первая операция в сверточном слое — это 3D свертка с несколькими различными фильтрами, параметры которых обучаются. Дальше к полученным картам признаков применяется нелинейное преобразование (в общем случае это может быть любая дифференцируемая функция, например, ReLU: $f(x) = \max(0, x)$ [21]). После этого может производиться локальная нормализация данных:

$$y_{ijk} = x_{ijk}\left(\kappa + \alpha \sum_{t \in G(k)} x_{ijt}\right)^{-\beta}$$

Это просто L_2 нормализация некоторого подмножества карт признаков $G(k)$. И в конце производится субдискретизация с выбором максимума, либо среднего в окне (рис. 3.2).

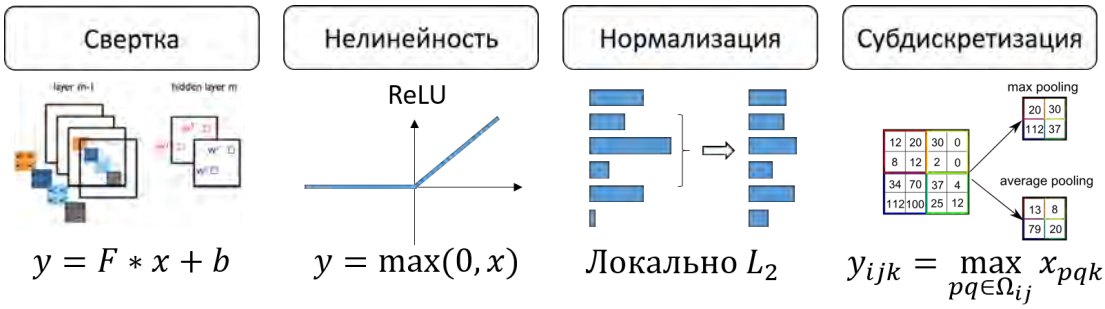

Рис. 3.2: Базовая структура сверточного слоя.

Обучение сверточных сетей производится с помощью стохастического градиентного спуска или его модификаций. Производные по параметрам сети вычисляются по правилу дифференцирования сложной функции с помощью хорошо известного метода обратного распространения ошибки.

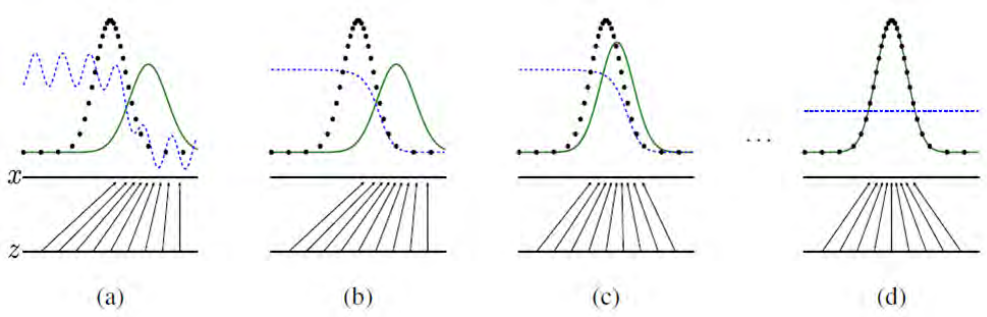

Рис. 3.3: Модельный пример того, как обучаются конкурирующие сети. Одновременно обновляется дискриминативная сеть D (синяя пунктирная линия) так, чтобы различать объекты из распределения данных p_{data} (черная точечная линия) и распределения, порождаемого генеративной сетью p_g (зеленая сплошная линия). Нижняя горизонтальная линия соответствует области из которой генерируются выборки из модельного распределения p_z. Горизонтальная линия выше — это область определения данных, куда отображаются модельные объекты с помощью сети G. На рисунке (a) изображена ситуация, когда модель уже практически обучилась: распределение p_g похоже на распределение p_{data}, а сеть D достаточно точно классифицирует объекты. На рисунке (b) изображена та же модель после выполнения внутреннего цикла алгоритма: сеть D стала оптимально классифицировать объекты. На рисунке (c) изображена модель после обновления параметров сети G: градиент сети D направляет сеть G в области, которые с высокой вероятностью будут классифицироваться как настоящие данные. На рисунке (d) изображена финальная модель после нескольких итераций: $p_g = p_{data}$, $D(x) = 0.5$.

3.2 Конкурирующие сети

Большинство подходов к решению задачи генерации фотореалистичных изображений используют для задания вероятностного распределения графические модели ([22], [23], [24]). Подобные модели зачастую оказываются неэффективными из-за необходимости проведения приближенных процедур вывода, а также использования Марковских цепей для генерации выборки. Недавно предложенная модель конкурирующих сетей [10] позволяет выучивать сложные распределения, а также генерировать выборки произвольного размера используя только вычислительно эффективный алгоритм обратного распространения ошибки для нейронных сетей.

На интуитивном уровне модель конкурирующих сетей работает следующим образом: одновременно обучаются две сети, генеративная сеть G и дискриминативная сеть D. Сеть G принимает на вход случайный шум и выдает на выходе изображение. Сеть D принимает на вход изображение и выдает на выходе вероятность того, что это изображение настоящее, а не было сгенерировано сетью G. При этом сеть D обучается как можно лучше разделять изображения на настоящие и нет, а обучение сети G заключается в максимизации вероятности сети D сделать ошибку. В качестве жизненной аналогии такого процесса обучения можно привести работу фальшивомонетчиков и полиции. Команда фальшивомонетчиков (генеративная модель) производит поддельные купюры, а полиция (дискриминативная модель) пытается отличать поддельные купюры от настоящих. Подобная конкуренция вынуждает каждую из сторон совершенствовать свои методы до тех пор, пока поддельные купюры не станут неотличимыми от настоящих.

Формально подобную модель можно записать в виде следующей минимакс игры:

$$G, D \leftarrow \min_G \max_D \left(\mathbb{E}_{x \sim p_{data}(x)} \left[\log D(x) \right] + \mathbb{E}_{z \sim p_z(z)} \left[\log \left(1 - D\left(G\left(z \right) \right) \right) \right] \right)$$

Здесь z — это некоторые модельные данные (случайный шум), $p_z(z)$ — это модельное распределение, выборки из которого подаются на вход сети G. x — объекты из распределения данных $p_{data}(x)$, из которого сеть G учится генерировать выборки. G, D — в общем

случае могут быть произвольными дифференцируемыми функциями, но мы будет рассматривать только ситуации сверточных нейронных сетей. Через p_g в дальнейшем будем обозначать распределение, задаваемое сетью G. Стоит отметить, что в ситуации, когда G и D произвольные функции, решению подобной минимакс задачи соответствует пара $D(x) = 0.5$, $p_g = p_{data}$, причем решать данную задачу можно с помощью алгоритма 1, с единственным условием, что сеть D на каждой итерации должна достигать своего точного максимума. На практике, конечно, никаких теоретических гарантий нет, так как мы параметризуем функции и не можем больше гарантировать нахождения точного решения. Поэтому, в частности, авторы статьи использовали гипер-параметр $K = 1$, так как это ведет к уменьшению вычислительных затрат. В качестве модельного распределения p_z обычно используется равномерное распределение $U[-1, 1]$. Модельная иллюстрация работы алгоритма 1 приведена на рис. 3.3.

Алгоритм 1 Стохастический градиентный спуск для обучения конкурирующих сетей. Шаг градиентного спуска может быть выполнен с помощью любого стандартного алгоритма. Авторы статьи в своих экспериментах использовали метод инерции [25]. В данной работе использовался метод Adam [26].

Вход: Общее количество итераций: T, количество итераций для сети D: K, модельное распределение $p_z(z)$

Выход: Обученные параметры нейронных сетей θ_d, θ_g

1: **Цикл** $t = 1..T$ **выполнять**

2: **Цикл** $k = 1..K$ **выполнять**

3: сгенерировать выборку $\{z_i\}_{i=1}^n$ из модельного распределения $p_z(z)$

4: сгенерировать выборку $\{x_i\}_{i=1}^n$ из распределения данных $p_{data}(x)$

5: обновить сеть D, сделав шаг по стохастическому градиенту:

$$\nabla_{\theta_d} \frac{1}{n} \sum_{i=1}^{n} \left[\log D(x_i; \theta_d) + \log\left(1 - D(G(z_i; \theta_g); \theta_d)\right) \right]$$

6: **Конец цикла**

7: сгенерировать выборку $\{z_i\}_{i=1}^n$ из модельного распределения $p_z(z)$

8: обновить сеть G, сделав шаг по стохастическому антиградиенту:

$$-\nabla_{\theta_g} \frac{1}{n} \sum_{i=1}^{n} \left[\log\left(1 - D\left(G(z_i; \theta_g); \theta_d\right)\right) \right]$$

9: **Конец цикла**

Модель конкурирующих сетей на сегодняшний день является одной из лучших моделей для генерации фотореалистичных изображений (пример изображений, полученных с помощью этой модели можно увидеть на рис. 3.4). Стоит отметить, что обучение конкурирующих сетей достаточно сложный и нестабильный процесс, который становится тем сложнее и нестабильнее, чем более глубокие сети используются в модели. Большое внимание при обучении стоит уделять синхронизации сетей D и G. Если сеть D будет недостаточно близка к оптимуму, то распределение, порождаемое сетью G может начать смещаться в произвольную сторону, необязательно соответствующую распределению p_{data}. Если же сеть D будет обучена слишком хорошо, то она будет практически на всех объектах выдавать значения вероятностей близкие к 0 или к 1, что приведет к очень маленьким значениям градиентов и, как следствие, значительному уменьшению скорости обучения сети G.

Эти проблемы были частично решены Дентоном, Чинталой и др. [27], которые предложили использовать Лапласовскую пирамиду конкурирующих сетей, что позволило, не изменяя сложности каждой отдельной модели генерировать фотореалистичные картинки высокого разрешения. Несколько позже, Радфольд, Метц и др. [28] предложили несколько важных практических рекомендаций, которые позволили сильно стабилизировать конкурирующее обучение и сделали возможным обучение более глубоких моделей.

Рис. 3.4: Примеры изображений, полученных с помощью модели конкурирующих сетей, реализованной в рамках данной работы.

Перечислим здесь эти рекомендации, так как в данной работе конкурирующее обучение было реализовано с использованием большинства из них:

- Заменить все операции субдискретизации на операции свертки с соответствующим шагом (дробным для генеративной сети).
- Использовать батч-нормализацию [29] как в дискриминативной, так и в генеративной сети.
- Не использовать полносвязные слои.
- Использовать функцию активации ReLU [21] в генеративной сети (за исключением последнего слоя, который использует Tanh).
- Использовать функцию активации LeekyReLU [30] для всех слоев дискриминативной сети.

4 Исследование применимости конкурирующих сетей для решения задачи устранения размытости изображений

Опишем еще раз основные идеи, мотивирующие использование конкурирующих сетей для решения задачи устранения размытости изображений. Анализ подходов к решению данной задачи показал насколько важным является внесение априорных знаний о виде фотореалистичных изображений в используемые модели. Тем не менее, современные нейросетевые методы явно эти знания не используют. Вкупе с тем фактом, что полученные с помощью минимизации L_2 нормы разницы исходных и восстановленных изображений результаты остаются несколько размытыми в сложных областях, становится ясно, что интеграция априорных знаний действительно может улучшить нейросетевые методы. Модель конкурирующих сетей естественным образом позволяет использовать эти знания, так как информация о том, как выглядят фотореалистичные изображения аккумулируется во время обучения в дискриминативной сети.

4.1 Метрики качества

На сегодняшний день наиболее распространенными метриками для оценки качества восстановленных изображений являются Peak Signal-to-Noise Ratio (PSNR) и Structural Similarity (SSIM) [31]:

$$PSNR(I_1, I_2) = 10\log_{10}\frac{L^2}{\|I_1 - I_2\|^2 / pq}$$

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

где I_1, I_2 — изображения размера $p \times q$, x, y — окна (обычно размера 8×8), прикладываемые к изображениям, L — максимальное значение яркости (обычно 255, либо 1), μ_x, μ_y, σ_x, σ_y, σ_{xy} — средние, дисперсии и ковариация изображений в окне, $c_1 = (0.01L)^2, c_2 = (0.03L)^2$ — константы. Заметим, что метрика SSIM определяется локально, поэтому, чтобы получить глобальную оценку похожести двух изображений используется усреднение SSIM по всем окнам. PSNR может принимать значения из диапазона $[0, +\infty]$, SSIM из диапазона $[-1, 1]$. Для обоих метрик верно, что чем больше их значение, тем более похожи сравниваемые изображения. Данные метрики обычно рассчитываются только для яркости изображений (канала Y цветовой модели YUV), хотя и могут быть легко обобщены на случай трехканальных изображений.

Во многих работах экспериментально было продемонстрировано, что как PSNR, так и SSIM плохо соответствуют человеческому восприятию ([32], [33], [34], [35], [36]). К тому же, минимизация PSNR эквивалентна минимизации L_2 нормы разницы исходных и восстановленных изображений, поэтому нейросетевые методы, минимизирующие попиксельную ошибку практически всегда будут показывать лучшее качество, с точки зрения PSNR, чем методы, обучающиеся минимизировать другие функции ошибки. Поэтому, целью данной работы было не получение наилучших значений с точки зрения PSNR или

SSIM, а именно получение визуально более качественных изображений. Количественное сравнение с помощью данных метрик приведено лишь для финального метода.

Опишем теперь использованные подходы для применения конкурирующих сетей для решению задачи устранения размытости изображений.

4.2 Стандартная модель конкурирующих сетей

Первый, наиболее естественный способ использования модели конкурирующих сетей заключается в замене генеративной сети на нейронную сеть, способную решать задачу устранения размытости. В данной работе использовалась сверточная сеть, аналогичная предложенной в [8] (эта модель описана в разделе 2.1 данной работы). Дискриминативная сеть работает стандартным образом: принимает на вход как настоящие изображения, так и изображения, восстановленные с помощью генеративной сети и обучается верно классифицировать все поданные объекты. Изначальная архитектура дискриминативной сети была заимствована из [28]. Итоговая модель изображена на рис. 4.1.

Перед тем, как обучать подобную модель, были сделаны несколько предположений о том, как должно проходить обучение. Дискриминативная сеть обучается как можно лучше различать настоящие и сгенерированные изображения. Так как генеративной сети на вход подаются размытые изображения, можно ожидать, что дискриминативная сеть обучится детектировать размытость (это будет наиболее простой способ различать сгенерированные и настоящие изображения). Это в свою очередь заставит генеративную сеть делать изображения более резкими и наиболее простым резким изображением, которое можно получить из размытого и будет настоящее исходное изображение, которое генеративная сеть обучится восстанавливать.

Для проверки сделанных предположений был поставлен ряд модельных экспериментов с простой моделью размытия:

$$y = x * k$$

где в качестве ядра свертки k использовалось ядро типа «диск» размером 3×3 или 5×5. В качестве обучающей выборки был использован набор данных «Imagenet» [37]. Каждое изображение исходного набора данных было обрезано так, что использовалась только центральная область размера 64×64, а значения пикселей были преобразованы в диапазон $[-1, 1]$. Генеративная сеть состояла из трех сверточных слоев с размерами фильтров, соответственно равными $1 \times 30 \times 32$, $30 \times 1 \times 32$, $5 \times 5 \times 128$ (последнее число равняется количеству карт признаков). Размер изображения все время оставался фиксированным за счет добавления фиктивных пикселей с нулевыми значениями на краях. После каждого сверточного слоя к картам признаков применялось нелинейное преобразование ReLU (за исключением последнего слоя, после которого использовался гиперболический тангенс, чтобы обеспечить корректные значения пикселей из диапазона $[-1, 1]$). Цветовые каналы обрабатывались независимо друг от друга. Заметим, что несмотря на то, что генеративная сеть работает с изображениями фиксированного размера, всегда можно разбить исходное изображение произвольного размера на пересекающиеся области размера 64×64 пикселей, обработать их поотдельности и произвести усреднение по пересекающимся областям. Дискриминативная сеть состояла из четырех сверточных слоев с размерами фильтров равными $5 \times 5 \times 64$, $5 \times 5 \times 128$, $5 \times 5 \times 256$, $5 \times 5 \times 512$. На каждом слое операции свертки применялись не ко всем позициям, а с шагом в 2, что позволило уменьшать размер изображений не используя операции субдискретизации (в некотором смысле подобный подход позволяет выучивать конкретную операцию субдискретизации во время обучения). В качестве функции активации использовалось преобразование LeekyReLU на всех слоях, кроме последнего, после которого использовалась сигмоида, чтобы преобразовать выход нейронной сети к диапазону $[0, 1]$ (сеть D выдает вероятности). Вход каждого слоя как генеративной, так и дискриминативной сети преобразовывался с помощью батч-нормализации.

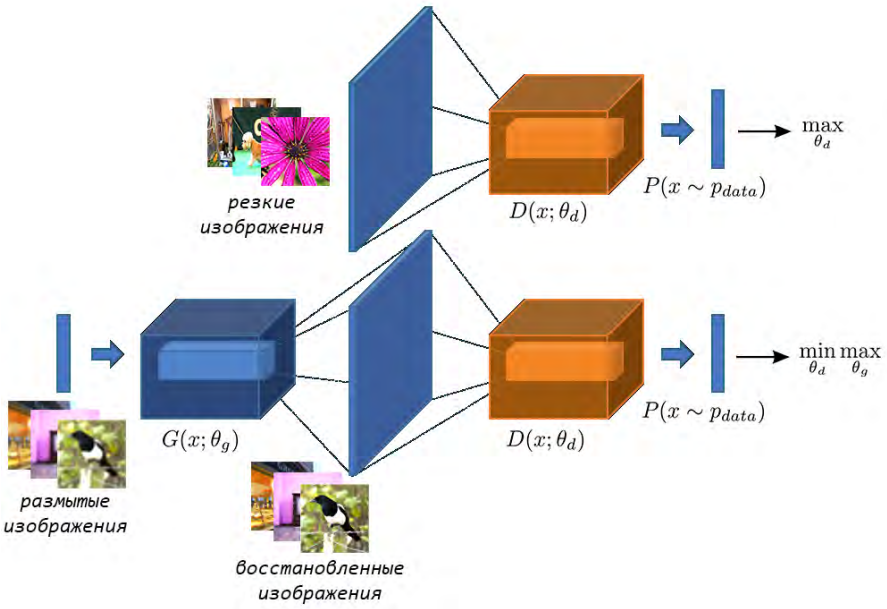

Рис. 4.1: Базовая модель конкурирующих сетей для решения задачи устранения размытости изображений.

Вся модель целиком обучалась с помощью алгоритма 1 (гипер-параметр $K = 1$, метод оптимизации: Adam).

Результаты, полученные с помощью данной модели можно увидеть на рис. 4.2 (a). Видно, что не смотря на то, что была выбрана наиболее простая модель размытия (при данной модели размытия задача, вообще говоря, может быть решена точно), генеративная сеть не восстанавливает исходные картинки. Тем не менее, восстановленные изображения выглядят достаточно реалистично. Полученные результаты предположительно означают, что гипотеза о том, что наиболее простое резкое изображение, которое можно получить из размытого будет исходным, оказалась неверной. Отметим, что это не означает, что модель конкурирующих сетей не обучается или неправильно работает в данной ситуации. Так как генеративная сеть не получает никакой информации о настоящих картинках, факт того, что она восстанавливает их неточно выглядит объяснимо и достоверно. Но перед тем, как делать вывод о том, что модель конкурирующих сетей в данном виде не применима к задаче устранения размытости, нами было исследовано несколько различных архитектур, использование которых могло сделать восстановление именно исходных изображений более предпочтительным для генеративной сети.

Первый проведенный нами эксперимент заключался в следующем: вместо того, чтобы показывать независимые изображения конкурирующим сетям, мы попробовали подавать на вход дискриминативной сети резкие изображения, являющиеся оригиналами для соответствующих восстановленных изображений из генеративной сети. При таком способе обучения дискриминативная сеть на каждой итерации может настраиваться на различия не между распределением данных и распределением сгенерированных изображений, а на различия между конкретными поданными изображениями. Стоит отметить, что достижение подобного эффекта не гарантируется, так как мы по прежнему явно не вносим информацию о соответствии изображений друг другу в функцию ошибки. Тем не менее, данный подход позволил несколько улучшить визуальное качество получаемых результатов, но в целом не решил исходной проблемы (рис. 4.2 (b)). Чтобы еще сильнее упростить задачу восстановления исходных изображений мы попробовали использовать остаточные слои [38] в генеративной сети. Остаточный слой — это сверточный слой, выход которого складывается с поданными на вход данными. Так как в нашем случае размытые и исходные изображения довольно похожи, использование остаточных слоев обосновано и может сделать задачу несколько проще, так как нейронной сети нужно будет выучить

(a) (b) (c)

Рис. 4.2: Результат работы первой предложенной модели конкурирующих сетей для задачи устранения размытости изображений. Верхний ряд: размытые изображения, средний ряд: восстановленные изображения, нижний ряд: исходные изображения. На рисунке (a) показан результат работы исходной модели. На рисунке (b) модели, с синхронизированным обучением сетей D и G (сеть D получает на вход оригиналы изображений, поданных сети G). На рисунке (c) показан результат работы модели с добавлением остаточного слоя.

лишь разницу между исходными и размытыми изображениями. В нашем случае вся генеративная сеть представляла из себя один остаточный слой, что эквивалентно добавлению размытого изображения, подаваемого на вход сети, к выходу последнего слоя. Использование подобной архитектуры позволило еще несколько улучшить визуальное качество, но исходная проблема по прежнему не была решена, что видно на рис. 4.2 (c).

В целом, по результатам проведенных экспериментов можно сделать вывод, что предложенная модель не подходит для решения задачи устранения размытости, так как изображения, восстанавливаемые генеративной сетью не всегда похожи на исходные. Заметим, что все эксперименты были проведены для предельно простого случая модели размытия и при усложнении задачи результаты могут только ухудшиться.

4.3 Комбинирование стандартного и конкурирующего обучения

Результаты экспериментов со стандартной моделью конкурирующих сетей (рис. 4.1) продемонстрировали необходимость введения в генеративную сеть информации о настоящих изображениях. Для того, чтобы это сделать была реализована модель, которая совмещает в себе стандартную нейронную сеть для решения задачи устранения размытости изображений и модель конкурирующих сетей (рис. 4.3). Отличие этой модели от предыдущей в том, что обучение генеративной сети происходит с использованием комбинированной функции ошибки, состоящей из взвешенной комбинации L_2 нормы разницы исходных и восстановленных изображений (стандартное обучение) и логарифма отклика сети D (конкурирующее обучение). Формально, оптимизируемая функция для модели может быть записана следующим образом:

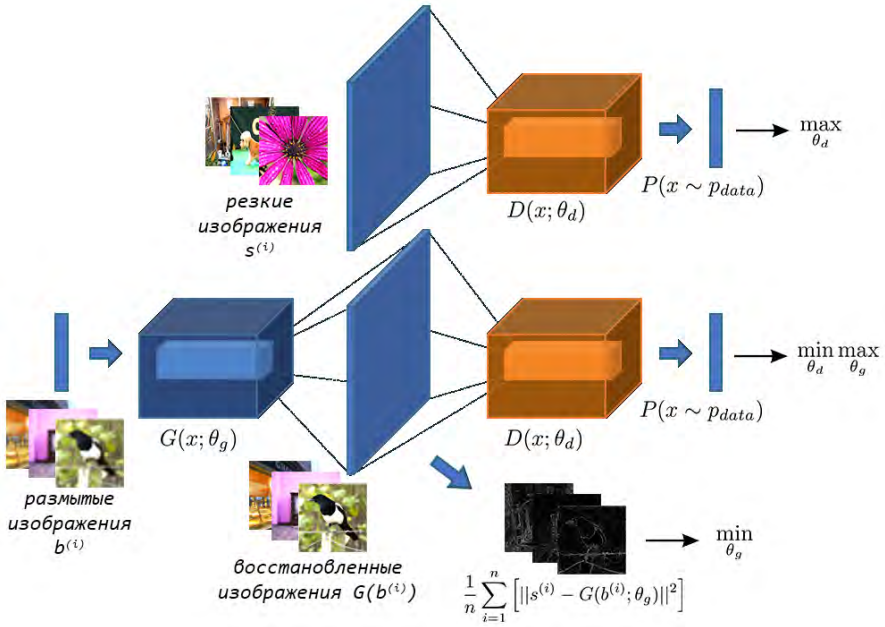

Рис. 4.3: Модель конкурирующих сетей для решения задачи устранения размытости изображений, комбинирующая стандартное и конкурирующее обучение.

$$\frac{1}{n}\sum_{i=1}^{n}\left[\log D\left(s^{(i)};\theta_d\right) + \log\left(1 - D(G(b^{(i)};\theta_g);\theta_d)\right)\right] \to \min_{\theta_d}$$

$$\frac{1}{n}\sum_{i=1}^{n}\left[\left\|s^{(i)} - G(b^{(i)};\theta_g)\right\|^2 - \lambda \log\left(D(G(b^{(i)};\theta_g);\theta_d\right)\right] \to \min_{\theta_g}$$

здесь $s^{(i)}, b^{(i)}$ — это исходные и размытые изображения соответственно, λ — гипер-параметр, регулирующий степень влияния конкурирующего обучения. Видно, что функция для обучение сети D осталась неизменной, а в функцию для обучения сети G добавилось слагаемое, отвечающее за L_2 норму разницы исходных и восстановленных изображений (также функция $\log(1 - D(x))$ была заменена на $-\log D(x)$, что соответствует той же точки минимума, но модуль градиента последней функции на практике значительно больше). Подобную схему обучения можно рассматривать как регуляризацию стандартной модели обучения нейронных сетей (в которой отсутствует сеть D и функция ошибки для сети G состоит только из первого слагаемого). Действительно, в процессе обучения сеть D будет выделять черты, по которым можно отличать восстановленные и настоящие изображения, и второе слагаемое в функции ошибки сети G как раз и будет поощрять более реалистичные с точки зрения сети D изображения. Если в целом сеть G будет генерировать размытые изображения, то второе слагаемое будет поощрять более резкие изображения. Если сеть G будет генерировать зашумленные изображения, то второе слагаемое будет поощрять незашумленные результаты, при условии, что сеть D сможет детектировать зашумленность и основываясь на этом классифицировать изображения.

Так как исходная задача некорректно поставлена, предполагалось, что добавление подобного регуляризатора позволит снизить круг возможных решений и сделает восстановленные картинки более резкими и реалистичными. В экспериментах с данной моделью использовалась довольно сложная модель размытия:

$$y = \phi(x * k + n)$$

где в качестве ядра свертки k использовалось ядро типа «диск» размером 15×15, к изображению добавлялся Гауссовский шум с параметром $\sigma \approx 0.02$ и после этого использовалось неравномерное затухание яркости (яркость изображения умножалась на 1.3 и к каждому пикселю применялась функция $c(x) = \max(c, 255)$), а также применялась JPEG-компрессия.

В рамках данного эксперимента было использовано несколько различных архитектур генеративной и дискриминативной сетей. Среди исследованных параметров были:

- Размер входного изображения (64, либо 128 пикселей, одноканальное изображение, либо трехканальное);
- Количество слоев сетей (от 3 до 5);
- Количество карт признаков и различные комбинации фильтров в каждом из слоев (3×3, 5×5 или 9×9, а также одномерные фильтры размеров 30 или 80 для первых слоев генеративной сети);
- Различные функции активации для внутренних слоев сетей (ReLU, LeakyReLU, Sigmoid);
- Различные функции активации для последнего слоя генеративной сети (Tanh, Sigmoid)[1];
- Использование батч-нормализации в обоих сетях;
- Использование остаточного слоя в генеративной сети;
- Различные значения гипер-параметров λ (0.1, 0.01, 0.001, 0.0001, 0.00001) и K (1, 2, 3, 9)[2];
- Различные методы оптимизации (стохастический градиентный спуск с использованием инерции, метод Adam, метод RMSProp [39]) и комбинации их параметров.

Заметим, что перебор был осуществлен не по всем возможным комбинациям описанных параметров, а лишь по некоторому подмножеству. Каждый раз изменялся один из параметров и оценивалось общее влияние на работу сетей и если оно не было положительным, то значение параметра оставалось прежним. В целом, данный поиск оказался оправданным, так как итоговая архитектура демонстрирует значительно лучшее визуальное качество, чем изначальная (рис. 4.4).

Итоговая модель работает с трехканальными изображениями размера 128×128, использует метод Adam (стохастический градиент каждый раз подсчитывался по 100 изображениям) с параметрами $\alpha = 0.001$[3], $\beta_1 = 0.9, \beta_2 = 0.999$, не использует остаточные слои, а также батч-нормализацию ни в генеративной, ни в дискриминативной моделях. Функции активации используются такие же, как и в предыдущей модели, генеративная сеть работает с фильтрами размеров $80 \times 1 \times 32$ и $1 \times 80 \times 32$, после которых идут еще два сверточных слоя с фильтрами размера $5 \times 5 \times 128$. В дискриминативной сети используется на один слой больше, чем в предыдущей модели с фильтрами $5 \times 5 \times 1024$. Параметры конкурирующего обучения (λ, K) часто выбирались различными и иногда менялись прямо в процессе обучения, чтобы достичь большей стабилизации сетей друг с другом. Также отметим, что основные результаты были получены путем дообучения генеративной сети, предобученной с параметром $\lambda = 0$. На рис. 4.5 представлены результаты работы модели, обученной подобным образом.

[1] При использовании функции Sigmoid все изображения нормировались в интервал $[0, 1]$, при использовании Tanh в $[-1, 1]$.

[2] Были также опробованы ситуации, когда для дискриминативной сети выполнялся один шаг по градиентному спуску, а затем несколько раз обновлялась генеративная модель.

[3] шаг градиентного спуска α уменьшался в 10 раз каждые 1000 итераций

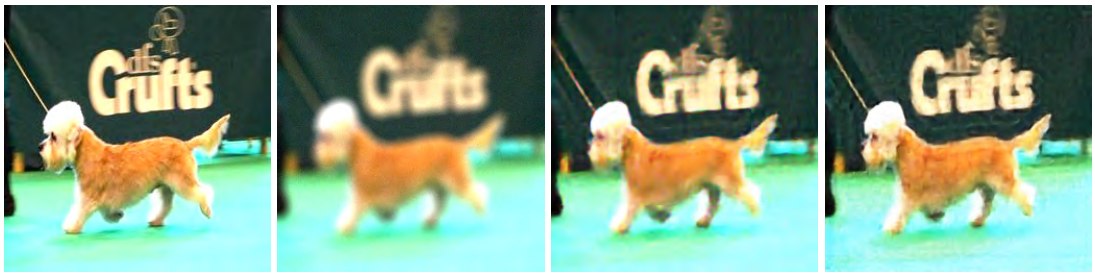

Рис. 4.4: Сравнение изначальной и итоговой архитектуры генеративной сети. Слева-направо: исходное изображение, размытое изображение, восстановленное с помощью изначальной архитектуры, восстановленное с помощью итоговой архитектуры. В данном случае для лучшего сравнения использовался параметр $\lambda = 0$ (отсутствие конкурирующего обучения), так как это делает модель значительно стабильней. Видно, что изображения, полученные с помощью итоговой архитектуры значительно более резкие.

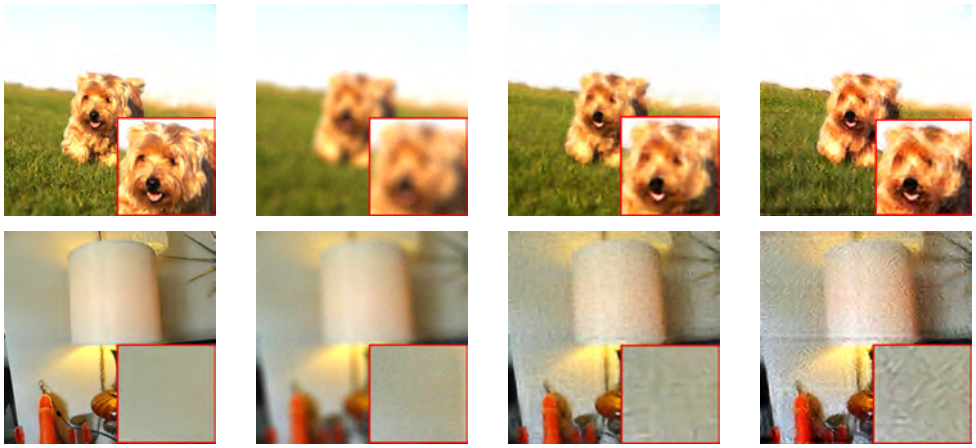

Рис. 4.5: Результат работы второй предложенной модели. Слева-направо: исходное изображение, размытое изображение, восстановленное без использования конкурирующего обучения ($\lambda = 0$), восстановленная с использованием конкурирующего обучения.

Можно обратить внимание на несколько интересных результатов. Во-первых, видно, что изображения действительно стали более резкими по сравнению с моделью, не использующей конкурирующее обучение. Но резкость на изображениях увеличивается за счет нанесения на них небольших штрихов, которые во многих случаях выглядят как случайный шум. Это наблюдение вполне объяснимо: дискриминативная сеть обучается детектировать размытость и вынуждает генеративную сеть восстанавливать более резкие изображения. Но наиболее простым способом «обмануть» дискриминативную сеть как раз и будет нанесение небольших резких штрихов равномерно на все изображение. Естественно, дискриминативная сеть обучается детектировать подобные зашумленные области, но это вынуждает генеративную сеть снова восстанавливать размытые изображения. Также, размытость поощряет слагаемое, отвечающее L_2 норме, которое в случае резких изображений начинает вносить больший вклад в функцию ошибки (так как L_2 норма зашумленных изображений больше, чем у несколько размытых). В итоге, в процессе обучения наблюдается осцилляция резкости изображений: генеративная сеть в начале генерирует размытые изображения, затем делает их более резкими за счет добавления шума, затем делает опять размытыми и т.д. (рис. 4.6).

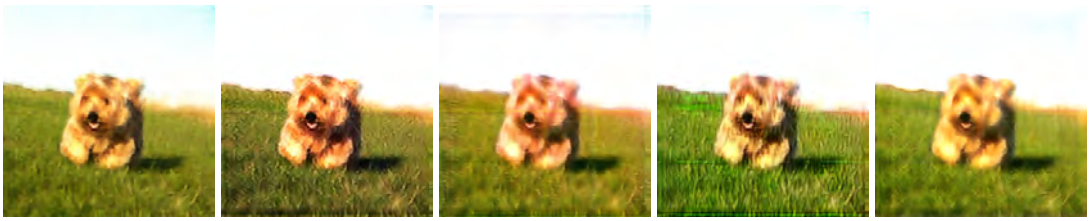

Рис. 4.6: Пример осцилляции резкости при обучении конкурирующих сетей с помощью второго подхода. Левое изображение получено с помощью предобученной модели без использования конкурирующего обучения. Каждое следующее изображение получено через несколько итераций комбинированного обучения. Видно, что резкость на изображениях действительно начинает осциллировать.

4.4 Использование скрытых представлений, формируемых дискриминативной сетью в функции ошибки

Анализируя полученные результаты можно сделать несколько выводов. Во-первых, комбинация стандартного и конкурирующего обучения для нейронных сетей действительно помогает устранить недостатки первого подхода и генеративная сеть начинает восстанавливать именно исходные изображения, а не некоторые фотореалистичные. Во-вторых, экспериментально было показано, что предложенный способ комбинации двух методов обучения неэффективен, так как резкость повышается равномерно, а структурные особенности изображений игнорируются. Чтобы исправить озвученную проблему нами был разработан принципиально другой подход к комбинации стандартного и конкурирующего обучения.

В процессе обучения на последних слоях дискриминативной сети формируются скрытые представления подаваемых ей на вход изображений. Данные скрытые представления содержат в себе всю информацию, используемую дискриминативной сетью для классификации изображений на настоящие и сгенерированные. Если сеть D детектирует размытость или зашумленность на изображениях, то эта информация кодируется в активациях нейронов на последних скрытых слоях. Соответственно, если у двух изображений формируются похожие скрытые представления, то это будет означать, что данные изображения похожи, с точки зрения сети D (например, оба размытые или оба резкие). Если потребовать, чтобы восстановленные сетью G изображения были близки к исходным не только в смысле по-пиксельного расстояния, но и в смысле расстояния между скрытыми представлениями, то полученные изображения должны стать более резкими, оставаясь при этом похожими на исходные. Формально, функцию ошибки в данном подходе можно записать следующим образом:

$$\frac{1}{n}\sum_{i=1}^{n}\left[\log D\left(s^{(i)};\theta_d\right)+\log\left(1-D(G(b^{(i)};\theta_g);\theta_d)\right)\right]\to\min_{\theta_d}$$

$$\frac{1}{n}\sum_{i=1}^{n}\left[\left\|s^{(i)}-G(b^{(i)};\theta_g)\right\|^2-\lambda\left\|D_f(s^{(i)};\theta_d)-D_f(G(b^{(i)};\theta_g);\theta_d\right\|^2\right]\to\min_{\theta_g}$$

где $D_f(x)$ — вектор признаков, формируемый на одном из слоев дискриминативной сети (в приведенных результатах использовался последний слой сети D). То есть, вместо максимизации вероятности дискриминативной сети совершить ошибку, предлагается обучать генеративную сеть минимизировать различия между скрытыми представлениями исходных и восстановленных картинок, формируемыми дискриминативной сетью. Схема подобной модели изображена на рис. 4.7.

В такой постановке задачи часть функции ошибки сети G, отвечающая за конкурирующее обучение начинает учитывать не только разницу между распределениями настоящих и восстановленных картинок в целом, но и разницу между конкретными картинками,

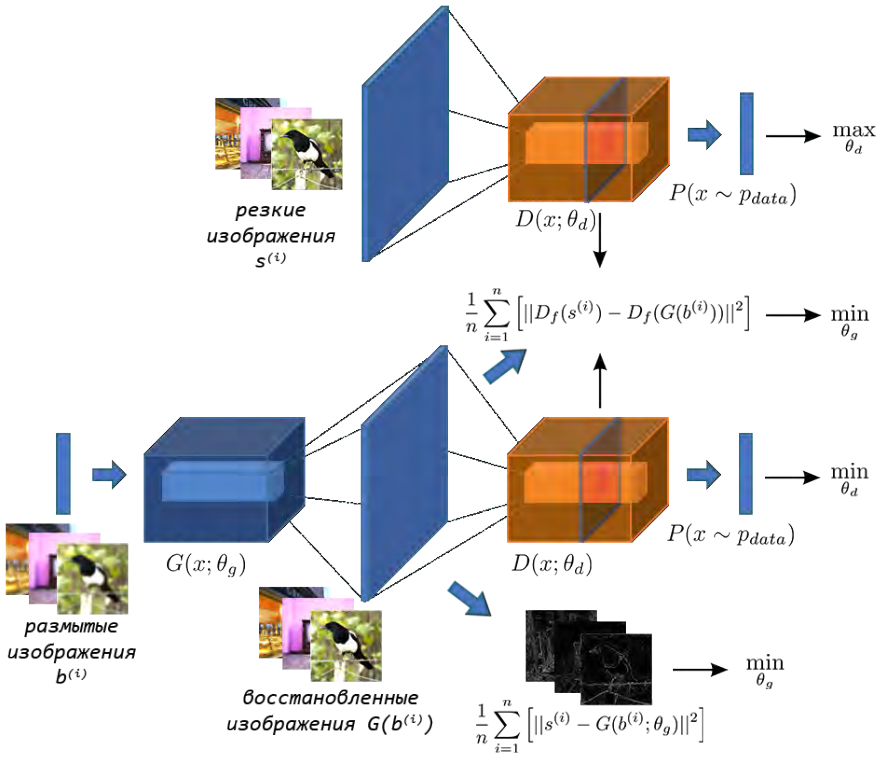

Рис. 4.7: Модель конкурирующих сетей для решения задачи устранения размытости изображений, использующая скрытые представления, формируемые дискриминативной сетью.

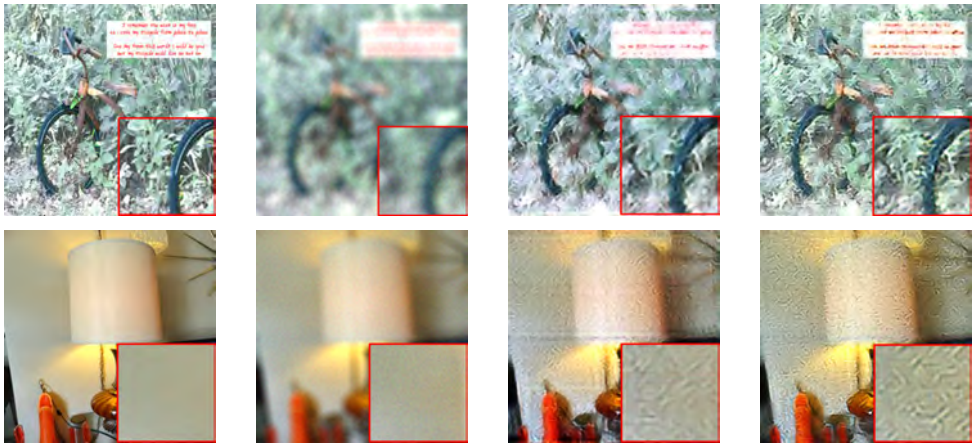

Рис. 4.8: Результат работы третьей предложенной модели. Слева-направо: исходное изображение, размытое изображение, восстановленное методом №2, восстановленное методом №3. Видно, что изображения становятся более резкими, но резкость все еще достигается за счет равномерного зашумления изображений.

что может помочь сделать результаты еще более качественными. На практике, однако, использование данной функции ошибки делает картинки еще более резкими, но не помогает устранить проблемы добавления шума или осцилляции резкости (рис. 4.8).

4.5 Минимизация функции ошибки, порождаемой глубинной нейронной сетью

Проведенные исследования показали, что конкурирующее обучение помогает сделать картинки более резкими, но делает их резкими в целом, игнорируя структуру изображений,

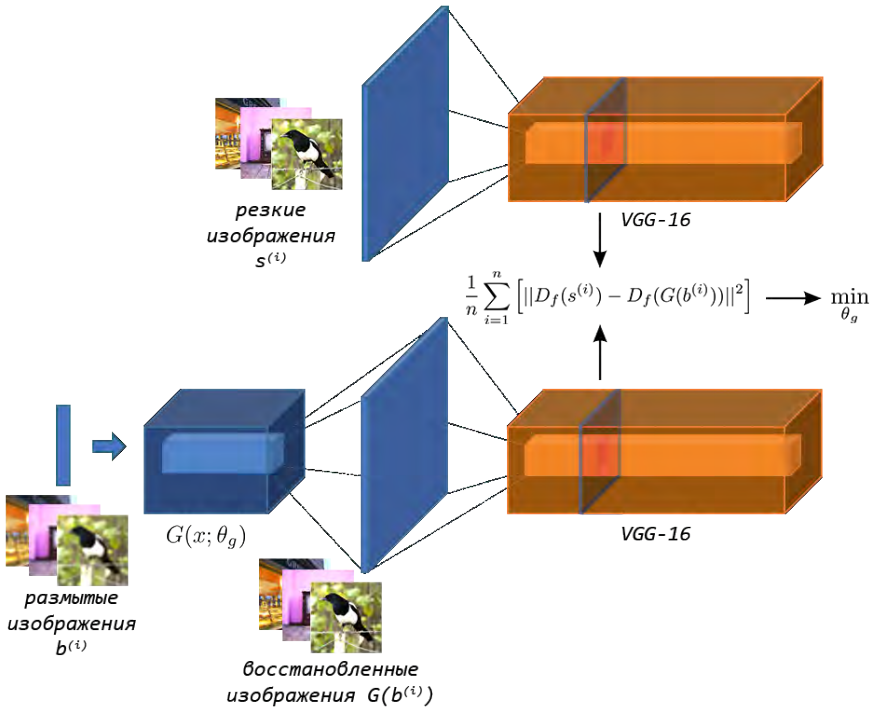

Рис. 4.9: Схема итогового метода для решения задачи устранения размытости изображений, основанного на минимизации L_2 нормы скрытых представлений, формируемых глубинной нейронной сетью, предобученной для классификации изображений.

одинаково добавляя новые границы как на резкие текстурные элементы (например, шерсть животных), где они необходимы, так и на монотонный фон (например, небо), где они являются шумом. Подобные наблюдения говорят о целесообразности внесения в модель явной информации о типе объектов, которые находятся на изображениях.

Для этого было предложено использовать глубинные сверточные нейронные сети семейства VGG, обученные классифицировать изображения на наборе данных Imagenet (во всех экспериментах использовалась нейронная сеть VGG-16 [13]). Для того, чтобы упростить задачу мы полностью убрали из модели конкурирующее обучение. Генеративная сеть обучалась минимизировать разницу между исходными и восстановленными изображениями, но не в смысле по-пиксельной L_2 нормы, а в смысле разницы между внутренними представлениями, формируемыми на первых слоях нейронной сети VGG-16. Идейно данный метод схож с методом обучения, сформулированным в предыдущем пункте за исключением того, что используется гораздо более глубокая модель, предобученная на классификацию изображений (это означает, что она как раз сможет отделять небо от животных и будет сильнее наказывать генеративную сеть за размытые области во втором случае). Кроме того, эта модель фиксирована, а не обучается вместе с генеративной сетью. В приведенных экспериментах использовались карты признаков, формируемые после слоя relu2_2. Изначально предполагалось обучение комбинированной функции ошибки, состоящей из L_2 нормы изображений и L_2 нормы их скрытых представлений. Однако, экспериментально было показано, что использование стандартной L_2 нормы в данном случае избыточно. Это означает, что признаки, формируемые на первых слоях глубинной сверточной нейронной сети еще достаточно локальные и, хоть и не содержат в себе явной информации о каждом пикселе исходного изображения, но позволяют восстановить его достаточно точно. Схема итоговой модели приведена на рис. 4.9.

Отметим также, что теоретически, в предложенных ранее подходах дискриминативная сеть также может научиться различать объекты на изображениях и не «наказывать» генеративную сеть за излишнюю размытость монотонного фона. Но на практике этого не

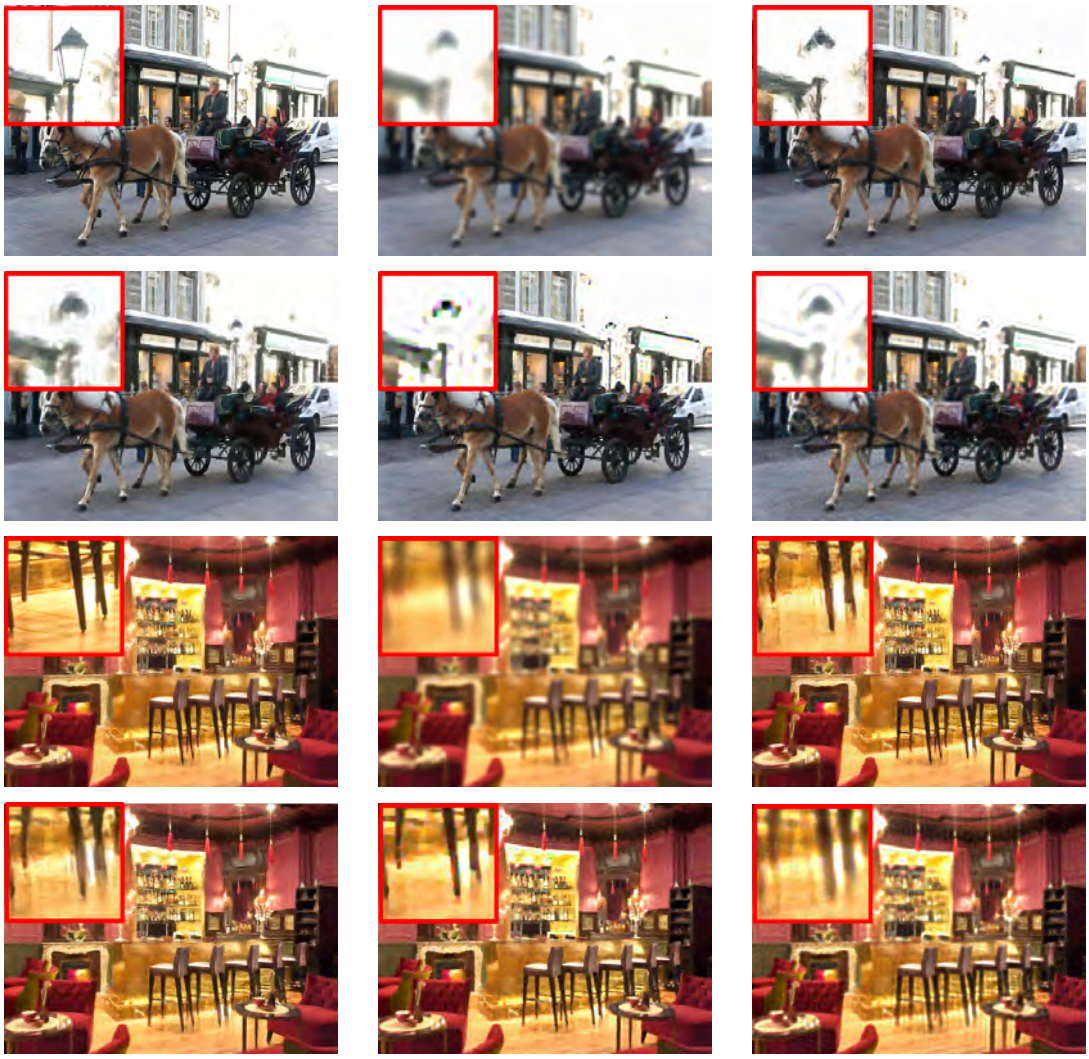

Рис. 4.10: Сравнение предложенного метода с современными алгоритмами решения задачи устранения размытости изображений. В верхнем ряду слева-направо: исходное изображение, размытое изображение, изображение, восстановленное предложенным методом. В нижнем ряду слева-направо: метод [8], метод [18] метод [19]. Видно, что предложенный метод восстанавливает более резкие изображения.

происходит, так как сеть D не была предобучена для классификации объектов на изображении и эта информация не является необходимой для правильного разделения изображений на настоящие и сгенерированные. Теоретически возможно использование в качестве сети D предобученной глубокой нейронной сети, хотя первичные эксперименты проведенные в данной работе показали, что это чересчур усложняет конкурирующее обучение и делает его нестабильным.

Экспериментально было продемонстрировано, что обученная предложенным в данном пункте образом генеративная модель во многих случаях демонстрирует значительно более резкие изображения, чем модели, обученные минимизировать L_2 норму. Было проведено экспериментальное сравнение данной модели с тремя современными методами для решения задачи устранения размытости изображений. Метод [8] — это наилучший на сегодняшний день нейросетевой метод, описанный в начале данной работы. Методы [18] и [19] — два метода, использующие сложные априорные распределения и явно моделирующие нелинейности вида неравномерного затухания яркости на изображениях. Полученные результаты приведены на рис. 4.10 и в таблице 4.1. Видно, что предложенный метод

	Blurred	Ху и др. [8]	Чо и др. [18]	Вайт и др. [19]	VGG-loss
Изображение 1 PSNR	20.81	24.31	**24.51**	23.17	23.61
Изображение 1 SSIM	0.23	0.42	**0.43**	0.38	0.37
Изображение 2 PSNR	20.74	24.57	**24.86**	22.98	23.81
Изображение 2 SSIM	0.27	**0.49**	0.49	0.41	0.42
В целом (по 30) PSNR	21.78	25.18	**25.47**	24.00	24.40
В целом (по 30) SSIM	0.23	0.41	**0.42**	0.35	0.36

Таблица 4.1: Количественное сравнение современных алгоритмов с предложенным методом.

действительно визуально лучше работает в сложных областях и вместо размытия небольших деталей делает их резкими. Однако, полученные значения метрик PSNR и SSIM существенно ниже, чем у других методов. Это еще раз подтверждает несоответствие данных метрик визуальному восприятию.

5 Заключение

В данной работе было проведено исследование по улучшению нейросетевых методов для решения задачи устранения размытости изображений за счет добавления в них априорной информации о виде фотореалистичных изображений. Для этого использовалась модель конкурирующих сетей. Были разработаны три различных способа обучения конкурирующих сетей для решения задачи устранения размытости, каждый следующий из которых проектировался так, чтобы устранять недостатки, присущие предыдущим методам. В итоге экспериментально было показано, что рассмотренные подходы действительно способны повысить резкость восстановленных нейросетевыми методами изображений, но новые границы часто выглядят как случайный шум и игнорируют структурные особенности изображений.

На основе проведенного исследования был разаработан и реализован алгоритм обучения нейронных сетей для решения задачи устранения размытости изображений за счет минимизации функции ошибки, порождаемой глубинной сверточной нейронной сетью, предобученной для решения задачи классификации изображений. Было проведено сравнение предложенного подхода с современными методами устранения размытости и продемонстрировано, что визуальное качество разработанного алгоритма во многих случаях выше, чем у другим методов.

Отметим, что возможные способы применения конкурирующих сетей для решения задачи устранения размытости изображений не ограничиваются рассмотренными в данной работе. В качестве дальнейшего развития можно предложить изменение архитектуры дискриминативной сети так, чтобы ей на вход подавались сразу два изображения: исходное и восстановленние и на выходе возвращалась некоторая оценка их схожести. Кроме того, за счет настройки параметров конкурирующего обучения может быть возможно использование предобученной глубинной нейронной сети в качестве сети D, что также может улучшить качество восстановленных изображений.

Литература

[1] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, “Removing camera shake from a single photograph,” *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 787—794, 2006.

[2] N. Joshi, C. L. Zitnick, R. Szeliski, and D. J. Kriegman, “Image deblurring and denoising using color priors,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1550—1557.

[3] Y. Zhou and N. Komodakis, “A map-estimation framework for blind deblurring using high-level edge priors,” in *Computer Vision--ECCV 2014*. Springer, 2014, pp. 142—157.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278—2324, 1998.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097—1105.

[6] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431—3440.

[7] J. Kim, J. K. Lee, and K. M. Lee, “Accurate image super-resolution using very deep convolutional networks,” *arXiv preprint arXiv:1511.04587*, 2015.

[8] L. Xu, J. S. Ren, C. Liu, and J. Jia, “Deep convolutional neural network for image deconvolution,” in *Advances in Neural Information Processing Systems*, 2014, pp. 1790––1798.

[9] C. Schuler, H. Burger, S. Harmeling, and B. Scholkopf, “A machine learning approach for non-blind image deconvolution,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1067—1074.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems*, 2014, pp. 2672—2680.

[11] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, “Understanding and evaluating blind deconvolution algorithms,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1964—1971.

[12] D. Krishnan, T. Tay, and R. Fergus, “Blind deconvolution using a normalized sparsity measure,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 233—240.

[13] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[14] N. Wiener, *Extrapolation, interpolation, and smoothing of stationary time series*. MIT press Cambridge, MA, 1949, vol. 2.

[15] L. B. Lucy, “An iterative technique for the rectification of observed distributions,” *The astronomical journal*, vol. 79, p. 745, 1974.

[16] W. H. Richardson, "Bayesian-based iterative method of image restoration*," *JOSA*, vol. 62, no. 1, pp. 55—59, 1972.

[17] S. Harmeling, S. Sra, M. Hirsch, and B. Schölkopf, "Multiframe blind deconvolution, super-resolution, and saturation correction via incremental em," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2010, pp. 3313—3316.

[18] S. Cho, J. Wang, and S. Lee, "Handling outliers in non-blind image deconvolution," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 495––502.

[19] O. Whyte, J. Sivic, and A. Zisserman, "Deblurring shaken and partially saturated images," *International Journal of Computer Vision*, vol. 110, no. 2, pp. 185—201, 2014.

[20] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with bm3d?" in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2392—2399.

[21] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807—814.

[22] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527—1554, 2006.

[23] R. Salakhutdinov and G. E. Hinton, "Deep boltzmann machines," in *International conference on artificial intelligence and statistics*, 2009, pp. 448—455.

[24] S. A. Eslami, N. Heess, C. K. Williams, and J. Winn, "The shape boltzmann machine: a strong model of object shape," *International Journal of Computer Vision*, vol. 107, no. 2, pp. 155—176, 2014.

[25] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1—17, 1964.

[26] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[27] E. L. Denton, S. Chintala, R. Fergus *et al.*, "Deep generative image models using a laplacian pyramid of adversarial networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1486—1494.

[28] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[30] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, 2013, p. 1.

[31] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600—612, 2004.

[32] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," *arXiv preprint arXiv:1603.08155*, 2016.

[33] D. Kundu and B. L. Evans, "Full-reference visual quality assessment for synthetic images: A subjective study," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2374—2378.

[34] P. Hanhart, P. Korshunov, and T. Ebrahimi, “Benchmarking of quality metrics on ultra-high definition video sequences,” in *Digital Signal Processing (DSP), 2013 18th International Conference on*. IEEE, 2013, pp. 1—8.

[35] Z. Wang and A. C. Bovik, “Mean squared error: love it or leave it? a new look at signal fidelity measures,” *Signal Processing Magazine, IEEE*, vol. 26, no. 1, pp. 98—117, 2009.

[36] Q. Huynh-Thu and M. Ghanbari, “Scope of validity of psnr in image/video quality assessment,” *Electronics letters*, vol. 44, no. 13, pp. 800—801, 2008.

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248—255.

[38] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *arXiv preprint arXiv:1512.03385*, 2015.

[39] T. Tieleman and G. Hinton, “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,” *COURSERA: Neural Networks for Machine Learning*, vol. 4, p. 2, 2012.