

Слепое устранение размытия изображения с использованием многозадачной сверточной нейронной сети и пространственно-вариативной рекуррентной нейронной сети

В.В. Анисимовский¹, М.А. Пенкин¹, Е.А. Дорохов¹, А.М. Груздев¹, С.С. Завалишин¹, И.В. Курилин¹

¹ Исследовательский центр Самсунг, ул. Двинцев 12, Москва, 127018, Российская Федерация

Аннотация

Мы представляем новый метод слепого устранения размытия изображения, способный обрабатывать пространственно изменяющееся размытие, вызванное различными причинами, такими как дрожание камеры и движения объекта. Из-за пространственно-вариативной природы сложного размытия наш метод устранения размытия включает как пространственно-вариативные, так и пространственно-инвариантные компоненты. А именно, мы используем гибридную нейронную сеть, состоящую из сверточной нейронной сети и пространственно-вариативной рекуррентной нейронной сети, обученную сквозным образом. Рекуррентная нейронная сеть используется в качестве оператора деконволюции, применяемого к признаковым картам, извлеченным из входного изображения сверточной нейронной сетью, в то время как попиксельные веса рекуррентной нейронной сети также генерируются той же сверточной нейронной сетью, что заставляет ее решать обе задачи одновременно. Такой гибридный и многозадачный дизайн позволяет нам достичь как низкой вычислительной сложности, так и высокого качества устранения размытия. Количественные и качественные оценки на широко используемом наборе данных GoPro демонстрируют, что предлагаемый метод обеспечивает хороший компромисс с точки зрения сложности и качества по сравнению с современными алгоритмами.

Ключевые слова

Слепое устранение размытия, глубокое обучение, рекуррентная нейронная сеть, пространственно-вариативное размытие, сверточная нейронная сеть.

Blind Image Deblurring Using Multi-Task Convolutional Neural Network and Spatially Variant Recurrent Neural Network

V.V. Anisimovskiy¹, M.A. Penkin¹, E.A. Dorokhov¹, A.M. Gruzdev¹, S.S. Zavalishin¹, I.V. Kurilin¹

¹Samsung R&D Institute Russia, 12, Dvintsev str., Moscow, Russia, 127018

Abstract

We present a novel method of blind image deblurring, capable of handling spatially variant blur caused by various reasons such as camera shake and object motions. Due to the spatially variant nature of the complex blur, our deblurring method includes both spatially variant and spatially invariant components. Namely, we utilize hybrid neural network consisting of a convolutional

ГрафиКон 2023: 33-я Международная конференция по компьютерной графике и машинному зрению, 19-21 сентября 2023 г., Институт проблем управления им. В.А. Трапезникова Российской академии наук, г. Москва, Россия

EMAIL: vanisimovsky@gmail.com (В.В. Анисимовский); m.penkin@samsung.com (М.А. Пенкин); e.dorokhov@samsung.com (Е.А. Дорохов); gruzdev.a@samsung.com (А.М. Груздев); etti.nekov@gmail.com (С.С. Завалишин); i.kurilin@samsung.com (И.В. Курилин)

ORCID: 0000-0001-9134-1078 (В.В. Анисимовский); 0000-0002-8027-9333 (М.А. Пенкин); 0009-0006-8259-2354 (Е.А. Дорохов); 0009-0007-1501-4973 (А.М. Груздев); 0009-0006-8418-8777 (С.С. Завалишин); 0009-0002-9015-4005 (И.В. Курилин)



© 2023 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

neural network and a spatially variant recurrent neural network trained in end-to-end manner. Recurrent neural network is used as a deconvolution operator performed on feature maps extracted from the input image by the convolutional neural network, while per-pixel weights of the recurrent neural network are also produced by the same convolutional neural network, making it solve both tasks at the same time. Such hybrid and multi-tasking design allows us to achieve both low computational complexity and high deblurring quality. Quantitative and qualitative evaluations on the widely used GoPro dataset demonstrate that the proposed method provides a good trade-off in terms of complexity and quality against state-of-the-art algorithms.

Keywords

Blind image deblurring, deep learning, recurrent neural network, spatially variant blur, convolutional neural network.

1. Введение

Размытие при движении, вызванное дрожанием камеры и перемещением объектов, является распространенной проблемой в цифровой фотографии. Поэтому за последние десятилетия были предприняты значительные усилия для решения задачи устранения размытия изображения. Однако восстановить четкое изображение из размытого затруднительно, поскольку это некорректно поставленная задача. Многие классические алгоритмы устранения размытия предполагают, что размытие вызвано простыми движениями камеры, такими как перемещение и поворот. Такие алгоритмы плохо работают для динамических сцен, где размытие вызвано не только движением камеры, но и движущимися объектами, с чем особенно трудно справиться, когда размытые движущиеся объекты находятся на существенно различных расстояниях от камеры.

Другое семейство алгоритмов устранения размытия основано на методах сегментации изображения для получения областей, требующих различной обработки. Такие методы обычно страдают от граничных артефактов, поскольку они зависят от точной сегментации, что само по себе является сложной задачей. Кроме того, большинство классических алгоритмов устранения размытия представляют собой итеративные процессы, решающие невыпуклые задачи оптимизации, что делает их весьма сложными в вычислительном отношении.

В последние годы, для устранения размытия динамических сцен нашли применение сверточные нейронные сети (СНС). Использование глубоких СНС позволяет обучать прямое восстановление четкого изображения из размытого, опуская оценку ядра размытия. Однако, поскольку веса СНС пространственно инвариантны, многие методы на основе СНС плохо работают для сцен с пространственно-неоднородным ядром размытия. Другой проблемой таких подходов является требование большого рецептивного поля СНС для эффективного устранения размытия, что приводит к большому размеру модели СНС и высокой вычислительной сложности. Для решения этой проблемы были использованы гибридные нейронные архитектуры, сочетающие СНС и пространственно-вариативные рекуррентные нейронные сети (РНС).

Наш метод относится к категории гибридных нейронных архитектур. Мы добиваемся улучшения базового метода, заставляя его СНС-компонент генерировать как весовые коэффициенты РНС, так и признаковые карты, вводимые в РНС. Наша цель - добиться разумного компромисса между вычислительной сложностью метода и качеством устранения размытия. Делая СНС-компонент нашего метода многозадачным, мы обеспечиваем такой компромисс, что мы демонстрируем, сравнивая наш метод с современными методами устранения размытия.

2. Похожие работы

Устранение размытия изображения - давняя задача в области обработки изображений и компьютерного зрения. Поскольку в этой работе мы фокусируемся на устранении размытия изображений вслепую на основе глубокого обучения, здесь мы кратко рассмотрим соответствующие методы.

Как в [1], так и в [2], СНС используется для оценки поля движения участка размытого изображения, которое впоследствии используется для обращения процесса размытия изображения. Другой метод [3] предложил глубокую мультимасштабную архитектуру СНС, обученную на парах размытых и четких участков изображения. Авторы этой работы также опубликовали свой крупномасштабный набор данных GoPro для обучения и оценки методов устранения размытия, который содержит четкие изображения, сделанные камерой GOPRO4 Hero Black со скоростью 240 кадров в секунду, и соответствующие размытые изображения, полученные из резких изображений путем усреднения различного количества последовательных кадров. В [4] был предложен более сложный, основанный на оптическом потоке, метод создания синтетического размытия наряду с еще большим набором данных и методом устранения размытия, основанным на СНС типа UNet, которая обрабатывает 5 последовательных размытых видеок кадров для создания центрального четкого кадра.

Тао и др. [5] предлагают масштабно-рекуррентную сеть, которая использует блоки сверточной долгой краткосрочной памяти (англ. Long short-term memory, LSTM), работающие в разных масштабах. Чен и др. [6] подходит к проблеме с позиций обучения без учителя, они обучают свою сеть устранения размытия используя повторное размытие изображений, из которых ранее было устранено размытие, для чего используется оптический поток, оцененный другой СНС. Купын и др. [7] используют генеративную состязательную сеть (ГСС) для решения задачи устранения размытия изображения, в то время как [8] добавляют пространственно-временную обработку к методу на основе ГСС, загружая в свою сеть 5 последовательных размытых кадров и используя 3D-свертки. Основываясь на успехе работы [9] в использовании обучаемых рекурсивных фильтров для обработки изображений, Чжан и др. [10] предлагают пространственно-вариативные рекуррентные нейронные сети для устранения размытия изображения. Работа [11] улучшает производительность модели ГСС из [7] за счет использования в генераторе облегченных вариантов базовых сетей.

Недавно работы [12] предложила адаптивную сеть с пространственно-временным фильтром, которая требует двух последовательных кадров для получения выходного сигнала с устраненным размытием. Чжан и др. [13] использовали иерархическую многопатчевую сеть с глубоким стекированием, в то время как [14] дополняют модель на основе ГСС двойными пропускающими соединениями в UNet-подобном генераторе. В [15] предложена масштабно-итеративная сеть увеличения масштаба, где остаточная плотная сеть и UNet объединены с масштабно-итеративной структурой для выполнения эффективного устранения размытия.

Наконец, в одной из недавних работ [16] предлагается многоступенчатая архитектура для прогрессивного восстановления изображения (включая устранение размытия). На ранних этапах используются кодеры-декодеры для извлечения мультимасштабных контекстуализированных признаков, в то время как последний этап работает с исходным разрешением изображения для получения пространственно точных выходных данных. Между каждым двумя этапами добавляется модуль внимания, который учится уточнять признаки одного этапа, прежде чем передавать их следующему этапу.

3. Описание метода

Сетевая структура нашего метода показана на рисунке 1. Мы опираемся на структуру, предложенную в [10]: сначала размытое изображение обрабатывается UNet-подобной многозадачной СНС, состоящей из кодера и декодера с пропускающими соединениями, которая выводит тензоры признаков и весовых коэффициентов, используемые в четырех последующих РНС, в то время как выходные данные последнего РНС обрабатываются небольшой СНС восстановления изображения для получения изображения без размытия. В качестве функции потерь при обучении используется L_2 -расстояние между выходным изображением и эталонным резким изображением.

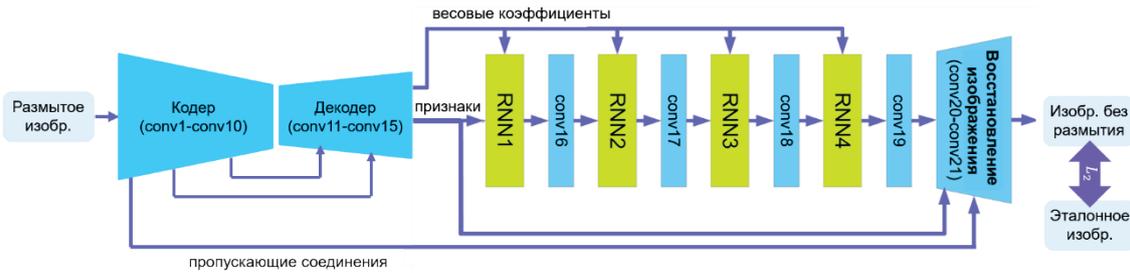


Рисунок 1 – Структура сети в нашем методе

Структура нашей модели отличается от структуры работы [10] тем, что мы не используем отдельную СНС для извлечения признаков. Вместо этого, как извлечение признаков, так и генерация весовых коэффициентов РНС выполняются одной глубокой многозадачной СНС. Мы утверждаем, что такой подход дает два преимущества по сравнению с моделью из [10]:

(а) наша модель работает быстрее из-за отсутствия отдельной СНС для извлечения признаков;

(б) тензор признаков, созданный многозадачной СНС, обеспечивает более богатое представление благодаря тому, что он выводится глубокой СНС, в то время как в [10] на вход РНС подаются признаки, созданные неглубокой СНС (состоящей всего из двух сверток).

Конфигурация многозадачной СНС идентична конфигурации СНС генерации весовых коэффициентов из [10] с единственным отличием в виде дополнительной свертки в конце СНС для получения тензора признаков, имеющего глубину 32 и те же ширину и высоту, что и у тензоров весовых коэффициентов. Другое отличие нашей модели от [10] заключается в СНС восстановления изображения: она состоит из двух сверток 3×3 (вместо 9×9 и 5×5 , как в [10]) и получает первое пропускающее соединение из тензора признаков, выводимого многозадачной СНС, в то время как второе пропускающее соединение исходит от первой свертки многозадачной СНС. Мы используем активацию с уткой LeakyReLU после всех сверток, за исключением двух последних сверток многозадачной СНС, производящих весовые коэффициенты РНС (там используется гиперболический тангенс) и тензор признаков (там активация не используется), а также кроме последней свертки СНС восстановления изображения. Подробное описание модели приведено на рисунке 2: все свертки имеют размер 3×3 , \downarrow означает пулинг с функцией максимума размером 2×2 , \uparrow означает билинейную интерполяцию разрешения в 2 раза, $[K_1:K_2]$ означает подмножество признаков признаковой карты, имеющих индексы от K_1 до K_2 и \oplus означает конкатенацию вдоль оси признаков. Каждый слой РНС принимает в качестве входных данных тензор признаков и тензор весовых коэффициентов (имеющих одинаковые размеры) и состоит из четырех проходов по входному тензору признаков: слева направо, справа налево, сверху вниз и снизу вверх, за которыми следует конкатенация четырех результирующих тензоров вдоль оси признаков.

Мы придерживаемся того же подхода к обучению, что и в [10]: используем набор данных GoPro из [3] и случайную нарезку патчей размером 128×128 . Набор данных GoPro содержит 3214 пар четких/размытых изображений, из которых 2103 пары были использованы для обучения, а остальные 1111 пар были использованы для теста. Четкие изображения получают с помощью камеры GOPRO4 Hero Black со скоростью 240 кадров в секунду и разрешением 1280×720 . Размытые изображения были получены путем усреднения различного количества (от 7 до 13) последовательных четких кадров для получения размытия различной интенсивности. Результирующие размытые изображения представляют собой очень разнообразный набор, включающий в себя различные ядра размытия и пейзажи.

Более компактная конструкция нашей модели позволила нам использовать при обучении пакеты большего размера - 32 по сравнению с 20 в [10]. Мы реализовали наш метод в TensorFlow [17] с пространственно-вариативной РНС, реализованной в соответствии с [9]. Многозадачная СНС была реализована с использованием слоев Keras (высокоуровневый интерфейс TensorFlow), в то время как для реализации пространственно-вариативной РНС мы использовали функцию `tf.keras.backend.rnn()` для запуска рекуррентного фильтра первого порядка по строкам/столбцам тензора признаков в прямом или обратном направлении, так что каждый блок РНС на Рисунке 1 содержит все четыре повторяющихся фильтра: слева направо,

справа налево, сверху вниз и снизу вверх, за которыми следует конкатенация четырех результирующих тензоров вдоль оси признаков. Обращаем внимание, что, в отличие от [10], мы не использовали никакого дополнительного увеличения обучающих данных или инициализации весовыми коэффициентами сети VGG16. Наша модель была обучена на графическом процессоре NVIDIA GeForce RTX 2080 Ti и процесс обучения сошелся после 4000 эпох.

	Слой	Вход	Выход
Кодер	conv1	$W \times H \times 3$	$W \times H \times 64$
	conv2	conv1	$W \times H \times 64$
	conv3	conv2↓	$\frac{W}{2} \times \frac{H}{2} \times 128$
	conv4	conv3	$\frac{W}{2} \times \frac{H}{2} \times 128$
	conv5	conv4↓	$\frac{W}{4} \times \frac{H}{4} \times 256$
	conv6	conv5	$\frac{W}{4} \times \frac{H}{4} \times 256$
	conv7	conv6	$\frac{W}{4} \times \frac{H}{4} \times 256$
	conv8	conv7↓	$\frac{W}{8} \times \frac{H}{8} \times 512$
	conv9	conv8	$\frac{W}{8} \times \frac{H}{8} \times 512$
	conv10	conv9	$\frac{W}{8} \times \frac{H}{8} \times 256$
Декодер	conv11	conv10↑ ⊕ conv6	$\frac{W}{4} \times \frac{H}{4} \times 128$
	conv12	conv11	$\frac{W}{4} \times \frac{H}{4} \times 128$
	conv13	conv12↑ ⊕ conv4	$\frac{W}{2} \times \frac{H}{2} \times 256$
	conv14	conv13	$\frac{W}{2} \times \frac{H}{2} \times 512$
	conv15	conv13	$\frac{W}{2} \times \frac{H}{2} \times 32$
Каскад РНС	RNN1	conv15, conv14[1:128]	$\frac{W}{2} \times \frac{H}{2} \times 128$
	conv16	RNN1	$\frac{W}{2} \times \frac{H}{2} \times 32$
	RNN2	conv16, conv14[129:256]	$\frac{W}{2} \times \frac{H}{2} \times 128$
	conv17	RNN2	$\frac{W}{2} \times \frac{H}{2} \times 32$
	RNN3	conv17, conv14[257:384]	$\frac{W}{2} \times \frac{H}{2} \times 128$
	conv18	RNN3	$\frac{W}{2} \times \frac{H}{2} \times 32$
	RNN4	conv18, conv14[385:512]	$\frac{W}{2} \times \frac{H}{2} \times 128$
	conv19	RNN4	$\frac{W}{2} \times \frac{H}{2} \times 32$
Восстановление изображения	conv20	conv19 ⊕ conv15	$\frac{W}{2} \times \frac{H}{2} \times 16$
	conv21	conv20↑ ⊕ conv1	$W \times H \times 3$

Рисунок 2 – Детальное описание нашей модели

4. Экспериментальные результаты

Мы оцениваем наш метод на тестовом подмножестве набора данных GoPro из [3] и сравниваем его с современными алгоритмами устранения размытия изображений. При измерении времени вывода этих алгоритмов на графическом процессоре NVIDIA GeForce RTX 2080 Ti мы заметили, что загрузка графического процессора была значительно ниже 100% и значительно различалась для разных методов. Таким образом, мы утверждаем, что измерения времени работы графического процессора (широко используемые для сравнения производительности методов устранения размытия) ненадежны в качестве показателя производительности для сравнения различных методов друг с другом, поэтому мы выполнили измерения времени вывода процессора на 6-ядерном процессоре AMD Ryzen 5 5600X. Поскольку загрузка процессора составляла 100% для всех методов, у нас больше уверенности в таком сравнении производительности.

Чтобы обеспечить справедливое сравнение, мы выбрали несколько современных методов устранения размытия из одного изображения, основанных на глубоком обучении без использования дополнительной информации (т.н. «слепые» методы устранения размытия). Мы не включили «неслепые» методы в сравнение, поскольку для них требуется информация о размытии, которая не всегда легкодоступна. Методы многокадрового устранения размытия также не включены, поскольку они неприменимы для устранения размытия из одного изображения и часто требуют больших вычислительных затрат.

В таблице 1 можно увидеть результаты количественной оценки и сравнения. Лучшие результаты выделены **жирным** шрифтом. Наши результаты выделены **красным** цветом. Из этой таблицы видно, что наш метод обеспечивает хороший компромисс между качеством устранения размытия (измеряемым с помощью PSNR/SSIM) и производительностью вычислений (измеряемой по времени работы процессора): все подходы, которые превосходят наш метод по PSNR/SSIM, имеют худшую производительность. Лучшим методом с точки зрения производительности является DeblurGAN-v2 (вариант с MobileNet) из [11], но его качество устранения размытия (по среднему PSNR) хуже на 1,39 дБ, чем наш результат. С другой стороны, лучшим методом с точки зрения качества устранения размытия является MPRNet из [16], но он более чем в 50 раз медленнее нашего метода. По сравнению с SVRNN из [10], который мы использовали в качестве базового, наш метод показывает то же процессорное время, в то время как PSNR улучшен на 0,37 дБ.

Таблица 1 – Количественная оценка качества и производительности по набору данных GoPro [3] с точки зрения PSNR, SSIM и времени вывода на процессоре для изображения 1280×720

Метод	PSNR↑, dB	SSIM↑	Время ЦПУ↓, сек
DeblurGAN-v2 (вариант с MobileNet) [11]	28.17	0.925	3.2
DeblurGAN [7]	28.70	0.927	7.8
SVRNN [10]	29.19	0.931	4.0
Наш метод (размер пакета: 20)	29.43	0.942	4.0
DeblurGAN-v2 (вариант с InceptionResNet-v2) [11]	29.55	0.934	10.4
Наш метод (размер пакета: 32)	29.56	0.943	4.0
SIUN [15]	30.22	0.904	4.5
Stack(4)-DMPHN [13]	31.20	0.945	138.1
MPRNet [16]	32.66	0.959	219.4

Чтобы проверить полезность нашего основного вклада, многозадачной СНС, мы дополнительно обучили нашу модель, используя размер пакета 20, чтобы соответствовать [10]. Из результатов этого эксперимента, приведенных в таблице 1, видно, что даже при меньшем размере пакета наш метод превосходит базовый метод из [10] как по PSNR, так и по SSIM, в то время как увеличение размера пакета до 32 (что стало возможным благодаря более компактному размеру нашей модели) улучшает результаты еще больше.

Примеры размытых изображений, обработанных нашим методом, приведены на рисунке 3. Можно видеть, что наш метод может устранять довольно сильное размытие и создает читаемый текст на изображениях.

5. Заключение

Мы предложили новый метод устранения размытия изображений, основанный на гибридной структуре, включающем многозадачную СНС, создающую как карты признаков, так и весовые коэффициенты для последующих РНС. Мы показали, что, используя наш новый многозадачный подход в гибридной структуре, сочетающей СНС и пространственно-вариативную РНС, можно добиться значительного улучшения качества выходного изображения без какого-либо увеличения вычислительной сложности. Обучив и оценив наш метод на широко используемом наборе данных GoPro, мы показали, что наш метод обеспечивает хороший компромисс с точки зрения качества устранения размытия и вычислительной производительности по сравнению с современными «слепыми» алгоритмами устранения размытия изображений. Ограничения нашего метода проистекают из конструкции пространственно-вариативной РНС: для каждого

такого блока РНС требуется свой собственный полноразмерный тензор весовых коэффициентов, поэтому мы были вынуждены ограничить количество пространственно-вариативных блоков РНС четырьмя, чтобы удержать объем памяти и вычислительную сложность на разумно низком уровне. Соответственно, наличие столь малого количества пространственно-вариативных блоков РНС ограничивает максимальный размер ядра размытия, для которого наша модель способна выполнять эффективное устранение размытия. Возможным направлением будущего развития нашего подхода может быть дальнейшее повышение эффективности пространственно-вариативной РНС при сохранении низкой вычислительной сложности.



Рисунок 3 – Примеры размытых изображений, обработанных нашим методом

6. СПИСОК ИСТОЧНИКОВ

- [1] Learning a convolutional neural network for non-uniform motion blur removal / J. Sun, W. Cao, Z. Xu, J. Ponce // In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015. pp. 769-777. doi:10.1109/CVPR.2015.7298677.
- [2] From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur / D. Gong et al. // In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017. pp. 2319-2328. doi:10.1109/CVPR.2017.405.
- [3] Deep multi-scale convolutional neural network for dynamic scene deblurring / S. Nah, T. Hyun Kim, K. Mu Lee // In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017. pp. 3883-3891. doi:10.1109/CVPR.2017.35.
- [4] Deep video deblurring for hand-held cameras / S. Su et al. // In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017. pp. 1279-1288. doi:10.1109/CVPR.2017.33.
- [5] Scale-recurrent network for deep image deblurring / X. Tao et al. // In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018. pp. 8174-8182. doi:10.1109/CVPR.2018.00853.
- [6] Reblur2deblur: Deblurring videos via self-supervised learning / H. Chen et al. // 2018 IEEE International Conference on Computational Photography (ICCP). 2018. pp. 1-9. doi:10.1109/ICCPHOT.2018.8368468.
- [7] Deblurgan: Blind motion deblurring using conditional adversarial networks / O. Kupyn et al. // In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018. pp. 8183-8192. doi:10.1109/CVPR.2018.00854.
- [8] Adversarial spatio-temporal learning for video deblurring / K. Zhang et al. // IEEE Transactions on Image Processing. 2018. № 28(1). pp. 291-301. doi:10.1109/TIP.2018.2867733.
- [9] Learning recursive filters for low-level vision via a hybrid neural network / Liu S., Pan J., Yang M.H. // In Computer Vision–ECCV 2016: 14th European Conference. 2016. Part IV. № 14. pp. 560-576. doi:10.1007/978-3-319-46493-0_34.
- [10] Dynamic scene deblurring using spatially variant recurrent neural networks / J. Zhang et al. // In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018. pp. 2521-2529. doi:10.1109/CVPR.2018.00267.
- [11] Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better / O. Kupyn, T. Martyniuk, J. Wu, Z. Wang // In Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019. pp. 8878-8887. doi:10.1109/ICCV.2019.00897.
- [12] Spatio-temporal filter adaptive network for video deblurring / S. Zhou et al. // In Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019. pp. 2482-2491. doi:10.1109/ICCV.2019.00257.
- [13] Deep stacked hierarchical multi-patch network for image deblurring / H. Zhang, Y. Dai, H. Li, P. Koniusz // In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019. pp. 5978-5986. doi:10.1109/CVPR.2019.00613.
- [14] Bi-skip: a motion deblurring network using self-paced learning / Y. Zhang et al. // arXiv preprint arXiv:1902.08915 (2019).
- [15] Scale-iterative upscaling network for image deblurring / Ye M., Lyu D., Chen G. // IEEE Access. 2020. № 8. pp. 18316-18325. doi:10.1109/ACCESS.2020.2967823.
- [16] Multi-stage progressive image restoration / S.W. Zamir et al. // In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021. pp. 14821-14831. doi:10.1109/CVPR46437.2021.01458.
- [17] Tensorflow: Large-scale machine learning on heterogeneous distributed systems / M. Abadi et al. // arXiv preprint arXiv:1603.04467 (2016).