

МОДЕЛИРОВАНИЕ АППЛИКАТИВНЫХ ПОМЕХ НА ИЗОБРАЖЕНИЯХ С ИСПОЛЬЗОВАНИЕМ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ

© 2022 А. А. Сирота, М. А. Дрюченко, М. С. Пузатых✉

*Воронежский государственный университет
Университетская пл., 1, 394018 Воронеж, Российская Федерация*

Аннотация. Моделирование различных специфических дефектов на изображениях имеет большое практическое значение при искусственном размножении данных в обучаемых алгоритмах распознавания, классификации и анализа изображений, при оценке устойчивости работы существующих алгоритмов обработки изображений, а также для тестирования специализированных алгоритмов, направленных на диагностику растровых искажений. Аппликативные помехи относятся к числу распространенных типов искажений цифровых изображений, затрудняющих работу систем технического зрения, действие которых проявляется в замещении отдельных участков полезного изображения фрагментами со случайной амплитудой, текстурой и формой.

В статье рассматривается задача моделирования аппликативных помех на изображениях с использованием современных алгоритмов статистической обработки и машинного обучения. Кратко описаны основные типы алгоритмов и методов моделирования аппликативных помех, включая непараметрические методы синтеза текстур, генерацию аппликативных помех как локальных областей закрытия исходных изображений, синтез текстур с использованием сверточных нейронных сетей, синтез аппликативных помех с помощью генеративно-сопоставительных нейронных сетей. Проведен сравнительный анализ статистического алгоритма генерации аппликативных помех как локальных областей закрытия с обучаемым алгоритмом генерации искаженных аппликативными помехами кадров, основанном на использовании GAN моделей. Для тестирования алгоритмов использовался набор данных TILDA Textile Texture Database. Предложен способ объективной оценки реалистичности получаемых аппликативных образований и искажений, основанный на применении глубоких сверточных классификаторов. Для решаемой задачи точность распознавания синтезированных аппликативных помех при использовании статистического алгоритма генерации оказалась чуть выше в сравнении с алгоритмом генерации помех на основе GAN моделей. Проанализированы результаты исследования степени реалистичности сгенерированных аппликативных помех на изображениях по сравнению с их естественными аналогами.

Ключевые слова: аппликативные помехи, генерация помех, нейронные сети, сверточные сети, GAN-сети.

ВВЕДЕНИЕ

В настоящее время цифровая обработка, анализ и распознавание изображений являются одними из наиболее интенсивно развивающихся направлений научных исследова-

ний. Это обусловлено широким распространением и использованием средств и систем машинного зрения, видеонаблюдения, моделирования и создания графических объектов в различных отраслях наук, а также появлением новых технологий в области обработки и анализа видеоданных различной природы.

Частой проблемой, встречающейся при работе с изображениями в задачах класси-

✉ Пузатых Михаил Сергеевич
e-mail: puzatyh.mihail@yandex.ru



Контент доступен под лицензией Creative Commons Attribution 4.0 License.
The content is available under Creative Commons Attribution 4.0 License.

фикации, фильтрации, сегментации, анализа и понимания сцен, является наличие помех и искажений различного типа, таких как локальные изменения уровней яркости, цифровые шумы, аппликативные помехи (АП), пространственные деформации фрагментов изображения и т. д. Такого рода помехи усложняют процесс получения требуемой информации и могут приводить к ошибкам при использовании различных алгоритмов обработки изображений [1]. Наиболее опасными на сегодняшний день считаются АП, проявляющиеся в возникновении локальных областей закрытия (затенения) или, локализованных в пределах анализируемой сцены, областей аномальных наблюдений (дефектов) на одном изображении или на серии изображений. Стоит отметить, что для многих практических задач исходные данные (цифровые фотореалистичные изображения, рентгеновские снимки, снимки, полученные с электронного микроскопа и т. п.) изначально допускают появление разного рода АП.

При обработке изображений в условиях АП следует выделить два подхода. Первый подход ориентирован на оценку качества исходных данных и их предварительную обработку с целью обнаружения и компенсации нежелательных искажений. Эффективность реализованных на этапе предварительной обработки алгоритмов зачастую определяет эффективность всей системы обработки изображений в целом. Второй подход предполагает непосредственное обучение алгоритмов анализа изображений, например, алгоритмов классификации, в условиях аппликативных и других помех на основе использования больших обучающих выборок, содержащих примеры изображений с различными типичными искажениями.

На данный момент одним из распространённых инструментов для обнаружения и сегментации дефектов на изображении являются нейросетевые технологии, а конкретнее — глубокие нейронные сети (НС) сверточного типа и их модификации. Такие сети могут использоваться и в рамках второго подхода, основанного на анализе изображений без специальной предварительной обра-

ботки. На сегодняшний день НС такого типа считаются одними из лучших алгоритмов по точности и скорости обнаружения объектов на анализируемой сцене. Сверточные НС хорошо справляются с большим количеством данных и достаточно эффективно обучаются на графических процессорах (GPU). Благодаря этим особенностям данный тип НС получил большое распространение.

Для обеспечения репрезентативности обучающих выборок при работе со сверточными НС в ситуациях, когда исходные наборы данных имеют ограниченный размер, могут применяться алгоритмы аугментации, реализующие искусственное искажение и размножение обучающих данных. Помимо стандартных искажающих преобразований, вроде поворотов и зашумления, процедуры аугментации могут включать моделирование АП, специфичных для определенных классов обрабатываемых тематических изображений.

Целью данного исследования является сравнительный анализ алгоритмов моделирования АП, которые могут быть использованы для аугментации (искусственного размножения данных, ИРД) при обучении нейросетевых классификаторов, действующих в условиях появления подобных аномальных наблюдений в пределах анализируемой сцены на одном или на серии изображений.

АНАЛИЗ РЕЗУЛЬТАТОВ ПРЕДШЕСТВУЮЩИХ РАБОТ

Непараметрические методы синтеза текстур. Основными методами непараметрического синтеза текстур являются методы, основанные на использовании патчей — участков изображений-образцов, содержащих требуемую текстуру. При использовании таких методов и реализующих их алгоритмов текстура синтезируемого изображения формируется путем комбинирования патчей [2, 3]. Так, например, предложенный в работе [3] алгоритм синтезирует новую текстуру попиксельно, начиная со случайного пикселя p . Для построения следующего пикселя выбирается патч $w(p)$, как квадрат со стороной w и с центром в p . Определяется $w_{best} = \arg \min_w d(w(p), w)$,

где $w(p)$ — квадратный патч с максимальной длиной стороны в синтезируемой текстуре, w — патчи из реального изображения, d — расстояние между патчами. Расстояние вычисляется только между построенными пикселями. Далее находятся прочие патчи w' , для которых $d(w'_{best}, w') < \varepsilon$, ε — заданный порог. На завершающем этапе находится значение пикселя p как среднего арифметического всех центральных p из найденных патчей w' и w'_{best} .

Описанный процесс повторяется до тех пор, пока не будет построена текстура требуемого размера. Стоит отметить, что при использовании непараметрических методов синтеза текстур возможны ситуации, при которых итоговое изображение не будет содержать требуемой текстуры. Это возможно если в процессе синтеза кадра не использовались патчи из интересующей «контекстной» части изображений-образцов. В зависимости от сложности и характера наполнения тематических изображений, искажаемых путем внесения АП, непараметрические алгоритмы синтеза текстур позволяют формировать как изображения самих помех, так и искаженных (аугментированных) кадров целиком.

Генерация аппликативных помех как локальных областей закрытия исходных изображений. Данный метод и реализующий его алгоритм [4] используют обобщенную статистическую модель генерации аппликативных помех, позволяющую создавать локальные закрытия произвольной площади и случайной формы с возможностью достижения эффекта полупрозрачности.

Искаженное изображение можно представить в виде

$$\tilde{I} = I + \eta w,$$

где I — исходное изображение, w — гауссовское случайное поле, η — случайное бинарное поле, определяющее локализации областей закрытия АП. В данном алгоритме под аппликативной помехой будет пониматься вся совокупности локальных областей закрытия (ЛОЗ) на изображении.

Весь процесс внесения аппликативных помех условно можно разбить на две части:

этап генерации условных центров ЛОЗ и этап формирования формы и текстуры, каждой отдельно взятой ЛОЗ.

Генерация условных центров ЛОЗ описывается пуассоновским распределением с некоторой заданной интенсивностью λ . По этой причине число таких условных центров ЛОЗ на изображении будет подчиняться распределению

$$P(S) = \frac{(\lambda V)^S}{S!} e^{(-\lambda V)},$$

где V — площадь изображения, $V = (N\Delta x)(N\Delta y)$, $\Delta x, \Delta y$ — дискреты задания координатной сетки. Параметр λ позволяет управлять количеством условных центров ЛОЗ, следовательно, и количеством самих ЛОЗ.

После генерации условных центров ЛОЗ происходит формирование конфигурации каждой отдельной ЛОЗ. Для каждого условного центра с использованием пуассоновского распределения с заданной интенсивностью α вычисляется значение n , определяющее площадь соответствующей ЛОЗ. Для уже выбранного условного центра ЛОЗ

$$P(n) = \frac{\alpha^{n-1}}{(n-1)!} e^{-\alpha}, \quad n \geq 1,$$

$$S = \Delta x \Delta y E[n], \quad E[n] = \sum_{n=1}^{\infty} n \frac{\alpha^{n-1}}{(n-1)!} e^{-\alpha},$$

где S — средняя площадь ЛОЗ.

После того, как определены площади ЛОЗ, проводится формирование формы каждой из них путем наращивания областей вокруг каждого условного центра. Генерация формы искажаемых областей происходит на бесконечной квадратной решетке, каждому узлу которой ставится в соответствие определенное значение прозрачности $T(x, y) \in [0, 1]$. Далее вводится дополнительный параметр $\tilde{T}(x, y) \in R$, определяющий концентрацию вещества физической среды, закрывающего изображение в данном узле решетки. Этот параметр однозначно связан с величиной $T(x, y)$. Данная величина зависит от соседних точек и определена случайным образом с условным распределением. То есть, если в определенной точке находится большое количество вещества физической среды, то считает-

ся, что и в соседних точках повышается ожидаемая концентрация вещества. В процедуре разрастания ЛОЗ используется маска расширения, центр которой помещается на уже искаженный пиксель и все пиксели, которые подпадают под данную маску могут считаться потенциальными кандидатами для искажения. Перед началом работы алгоритма выбирается условный центр помехи с координатами (i, j) — точка, относительно которой будет происходить генерация ЛОЗ, а также задаются начальные распределения и максимальная площадь ЛОЗ. В итоге значение случайно-бинарного поля $\eta(x, y)$ определяется как

$$\eta(x, y) = \begin{cases} 1, & p(\eta(x, y) = 1) = T(x, y) \\ 0, & p(\eta(x, y) = 0) = 1 - T(x, y) \end{cases}$$

Альтернативный вариант генерации ЛОЗ предусматривает использование градиентных шумов, таких как шум Перлина [5]. При генерации двумерного изображения шума Перлина рассматривается регулярная сетка, каждому узлу которой ставится в соответствие единичный вектор с заданным псевдослучайным образом углом наклона, называемый градиентом. Данные вектора указывают направление интерполяции при вычислении значений отдельных пикселей при построении изображения шума.

Бинарная маска J_B , содержащая связанные множества пикселей, соответствующие ЛОЗ, может быть получена в результате сегментации двумерного изображения градиентного шума J . Например, может использоваться глобальная сегментация с параметрически заданным порогом t :

$$J_B = T_t(J_S),$$

$$J_S = \frac{1}{2\pi\sigma^2} \sum_{u,v} e^{-\frac{(u^2+v^2)}{2\sigma^2}} J(x+u, y+v),$$

где J_S — сглаженное по Гауссу с заданным СКО изображение градиентного шума G . В зависимости от параметров сгенерированного градиентного шума (амплитуды, частоты, стойкости, числа октав), а также в зависимости от выбранного значения порога сегментации t на маске J_B может содержаться различное число стохастических ЛОЗ, характеризующихся различной площадью, компактностью,

регулярностью формы. Учитывая данные характеристики, а также требуемое число ЛОЗ на целевом искажаемом кадре, к маске J_B применяются процедуры фильтрации бинарных объектов, позволяющие оставить на J_B ограниченное число связанных областей пикселей соответствующих формируемым АП.

Итоговое искаженное изображение можно представить в виде

$$\tilde{I} = (I - J_B) + \mu(I - (255 - J_B)) + \nu\tilde{J},$$

где $\tilde{J} = F_{noise}(J - (255 - J_B), m_l, D_l)$ — изображение АП в градациях серого или в RGB (совпадает с глубиной цвета J) с добавлением к итоговому представлению гауссовского шума, параметры которого определяются на основе оцененных значений математического ожидания m_l и дисперсией D_l искажаемого кадра, μ, ν — коэффициенты прозрачности и яркости помех.

Синтез текстуры изображений с использованием сверточных нейронной сети.

В рамках данного подхода получается модель текстуры, параметризованная пространственно инвариантными представлениями, построенными на иерархической архитектуре слоев обработки сверточной НС. Как пример, в работе [6] используется сверточная НС VGG-19, обученная распознаванию объектов, представленных своими изображениями. Для генерации новой текстуры из исходного изображения извлекаются признаки различных размеров, после чего вычисляется пространственная сводная статистика, чтобы получить стационарное описание исходного изображения. Далее, выполняя градиентный спуск на случайном изображении, инициализированном белым шумом, находится новое изображение с аналогичным стационарным описанием.

Синтез аппликативных помех с помощью нейронных сетей класса GAN. Генеративно-состязательные сети (GAN) относятся к числу современных НС, позволяющих реализовать эффективные алгоритмы синтеза реалистичных изображений [7–9]. Данный класс НС построен на комбинации двух НС: генератор G , который синтезирует изображения, и дискриминатор D , которые производит

сравнение сгенерированного изображения с реальным изображением. Процесс обучения заключается в максимизации генератором вероятности ошибки дискриминатора. Обобщенная архитектура GAN сетей представлена на рис. 1.

Рассмотрим конкретную реализацию DefectGAN сети. Предполагается, что существует достаточное количество нормальных образцов (без дефектов) и ограниченное количество образцов дефектов. Основываясь на этом предлагается выполнить синтез дефектов используя преобразование изображения в изображение [10]. Имитируя процессы деформации и восстановления, можно использовать большое количество нормальных образцов для создания требуемых образцов дефектов.

На этапе обучения DefectGAN выполняет преобразование изображений с помощью генератора в два шага: $n \rightarrow d \rightarrow \tilde{n}$ и $d \rightarrow n \rightarrow \tilde{d}$, где $n \in R^{H \times W \times 3}$ обозначает нормальный образец, а $d \in R^{H \times W \times 3}$ обозначает дефект, $\tilde{n}, \tilde{d} \in R^{H \times W \times 3}$ — восстановленные образцы. Далее будет описан только цикл $n \rightarrow d \rightarrow \tilde{n}$. Генератор G использует архитектуру кодер-декодер. Общая структура генератора показана на рис. 2.

Данная архитектура соответствует широко используемым сетям преобразования изображений в изображения, которые сначала кодируют входное изображение с шагом 4, а затем декодируют его до исходного размера. Чтобы улучшить реалистичность и разнообразие результатов синтеза при генерации дефектов, используется пространственный и категориальный контроль. Архитектура D аналогична

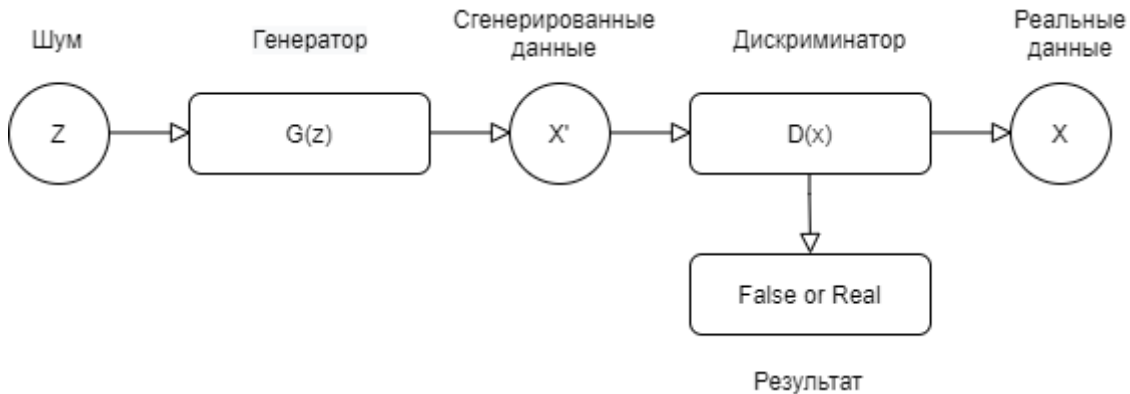


Рис. 1. Обобщенная архитектура GAN сетей [Fig. 1 Generalized architecture of GAN network]

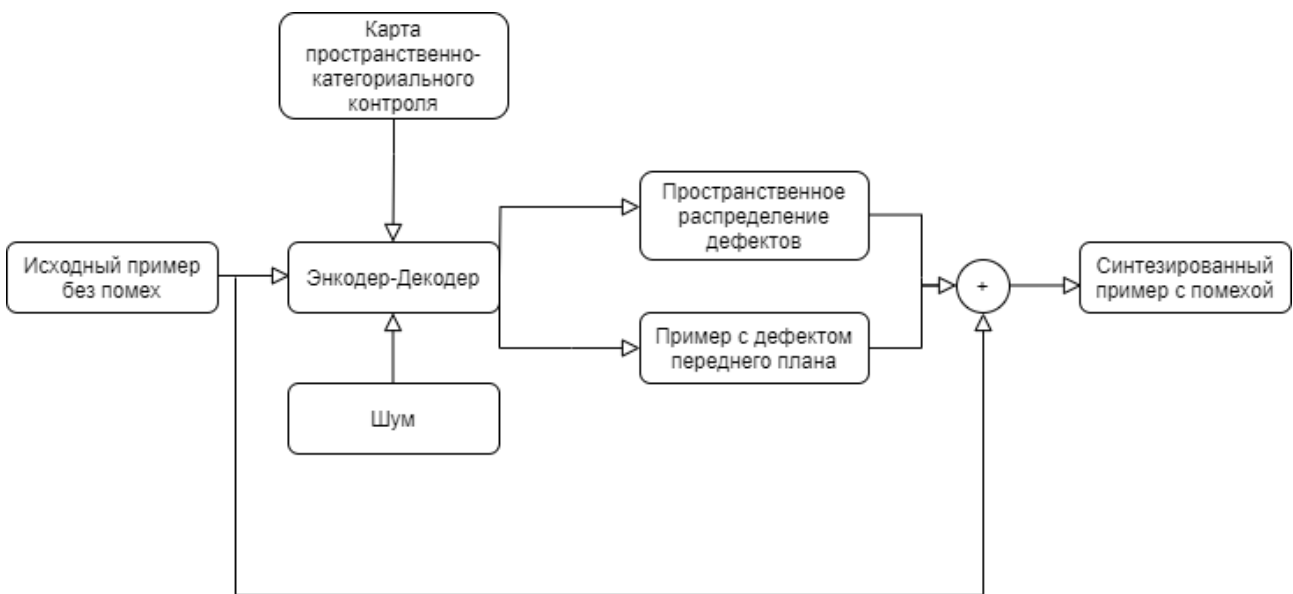


Рис. 2. Архитектура генератора сети DefectGAN [Fig. 2 Architecture of generator for DefectGAN network]

StarGAN [11], которая используется для распознавания поддельных выборок от реальных для прогнозирования категорий сгенерированных дефектов. Также для имитации дефектов, которые обладают сложными и нерегулярными шаблонами с чрезвычайно высокими стохастическими вариациями, используется адаптивная вставка шума в каждом блоке архитектуры кодер-декодер, который явно вводит гауссовский шум в карты характеристик после каждого сверточного блока. DefectGAN также отличается от существующих GAN тем, что последним этапом генерации является объединение двух слоев. В частности, в процессе искажения окончательные образцы дефектов генерируются путем добавления слоя переднего плана дефекта поверх предоставленных нормальных образцов. Вместо генерации синтезированного изображения напрямую, DefectGAN отдельно генерирует дефект переднего плана вместе с соответствующими картами пространственного распределения, а затем выполняет послойную композицию для создания конечного изображения с дефектом. Взяв нормальные образцы в качестве фона, модель неявно фокусируется на генерации дефектов, не учитывая генерацию фона. Эта функция предоставляет модели больше возможностей для создания более реалистичных образцов дефектов. Данный факт значительно улучшает переносимость дефектов, что означает, что предлагаемая модель способна генерировать новые образцы дефектов на различном фоне.

МЕТОДЫ И МАТЕРИАЛЫ

В ходе исследования нами рассматривались два метода синтеза АП. Первый алго-

ритм основан на рассмотренной выше модели генерации АП как локальных областей закрытия исходных изображений. Вторым методом основан на синтезе АП с помощью GAN НС, а именно архитектура DefectGAN. Данный выбор методов обуславливается их универсальностью и независимостью относительно данных, на которых требуется синтез АП. Для объективного анализа степени реалистичности полученных АП (дефектов) на изображении предлагается использовать классификатор в виде сверточной НС, которую предлагается обучить для распознавания наличия дефектов на изображении. При этом предлагается с помощью такой сети детектировать как реальные, так и искусственно синтезированные дефекты.

Исследование проведено на наборе данных TILDA Textile Texture Database, содержащего изображения тканей с различного рода искажениями, по своим проявлениям отвечающие влиянию АП в виде локализованных областей аномальных наблюдений. Примеры изображений и дефектов на них приведены ниже на рис. 3.

Описание классификатора для анализа степени реалистичности дефектов. Для анализа степени реалистичности генерируемых искусственных дефектов используется классификатор — сверточная НС, архитектура которой представлена на рис. 4. Здесь слой Conv2D является сверточным, в качестве функции активации используется функция ReLU, а далее идет слой субдискретизации MaxPooling2D. Данные слои можно выделить в отдельный блок, который в архитектуре повторяется 3 раза с разницей гиперпараметров

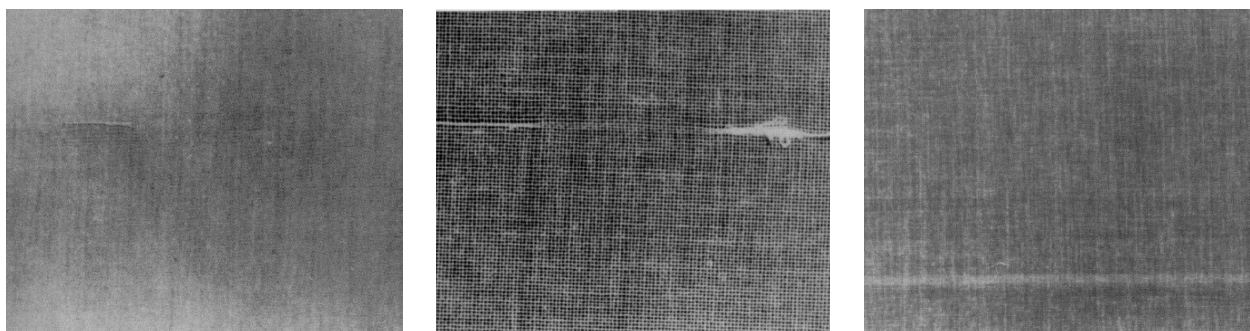


Рис. 3. Примеры исходных изображений
[Fig. 3 Examples of source images]

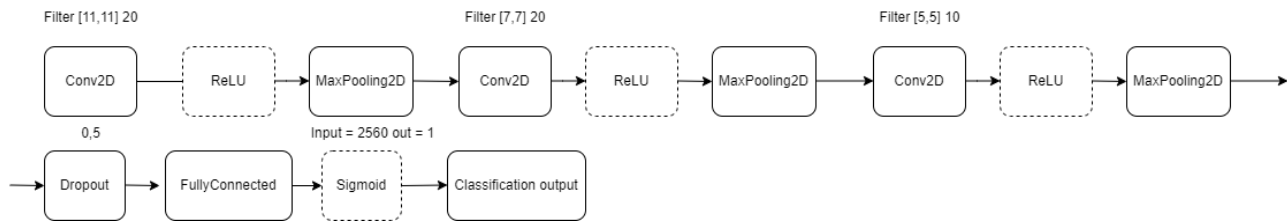


Рис. 4. Архитектура классификатора дефектов в виде сверточной нейронной сети
 [Fig. 4 Architecture of the defect classifier in the form of a convolutional neural network]

в сверточном слое. Для первого блока ядро свертки имеет размерность блоков 11×11 , для второго — 7×7 , для третьего 5×5 . В качестве инструмента для регуляризации в целях избежания переобучения сети была использована техника dropout (выключение из дальнейшего обучения некоторого процента нейронов, выбранных случайным образом). В данном случае процент не использующихся в дальнейшем обучении нейронов был равен 50 %. В модели используется полносвязный слой, имеющий 2560 входов и 1 выход. В качестве функции активации на выходе сети использовалась функция «сигмоид».

Формирование обучающей выборки.

Для сравнительного анализа эффективностей алгоритмов генерации АП было решено использовать два варианта формирования обучающих выборок. Выборки формировались из исходных реальных изображений и изображений с искусственно синтезированными АП. В выборке первого типа (выборка 90/10) 90 процентов обучающих данных составляли изображения с искусственными АП, а 10 процентов реальные изображения. В выборке второго типа (выборка 10/90) 10 процентов составляли изображения с синтезированными АП, а 90 процентов составляли реальные изображения.

Обучение проводилось по 10 эпохам, использовался $\text{minibatch} = 20$ фрагментам. Достигнутое значение ошибки валидации составило $\text{Err}_{\text{val}} = 0,89$. Объем обучающей выборки в первом варианте составил 1500 фрагментов, содержащих АП и 2500 фрагментов, не содержащих АП. Для второго варианта объем обучающей выборки был идентичным.

Подход для сравнения эффективности алгоритмов. Во время сравнительного анализа был использован следующий подход. Фор-

мировалось две выборки, описанные выше, на одной из них обучалась сверточная НС.

На следующем шаге данная обученная сверточная НС анализировала разбитое на блоки изображение. Для оценки эффективности были выделены следующие возможные варианты реагирования для нейронной сверточной сети. Корректно обнаруженным объектом считался тот случай, если нейронная сверточная сеть выделяла хотя бы один из блоков, который содержит АП, и обозначала его как дефектный. Пропущенным объектом считался тот случай, когда НС не выделяла ни один из блоков, на котором присутствовала АП.

Для сравнения использовались следующие общепринятые метрики: вероятности ошибок первого и второго рода. Под ошибкой первого рода понимается ситуация ложноположительного срабатывания, когда НС отмечала фрагмент изображения, не содержащего реального дефекта. Вероятность ошибки первого рода можно выразить как:

$$P_{\text{err}}^{(1)} = \frac{k}{K},$$

где k — число неверно обнаруженных фрагментов изображения, $K = W \times H \times L$, W — ширина блока, H — высота блока, L — количество кадров.

Ошибке второго рода соответствовала ситуация ложноотрицательного срабатывания, когда НС не отмечала фрагмент изображения, как дефектный, хотя на данном фрагменте присутствовала АП или ее часть. Вероятность ошибки второго рода можно представить в виде:

$$P_{\text{err}}^{(2)} = 1 - \frac{n}{N},$$

где n — количество обнаруженных фрагментов содержащих АП, N — общее число фрагментов, содержащих АП.

РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТА

Для синтезирования АП использовались два алгоритма: алгоритм генерации АП как ЛОЗ исходных изображений и алгоритм на основе GAN сети. Искаженные изображения разбивались на блоки размерностью 128×128 пикселей, которые подавались на вход алгоритма распознавания и классификации реалистичности дефектов.

С использованием статистического алгоритма (СА) генерировались АП с различной степенью прозрачности и площадью закрытия (в среднем не превышающие 20 процентов от размера анализируемого блока). Отметим, что при тестировании рассматривалась реализация СА, использующая двумерный градиентный шум Перлина для формирования ЛОЗ.

При обучении сети DefectGAN использовался оптимизатор Adam с $\beta_1 = 0,5$ и $\beta_2 = 0,999$, обновление генератора происходило каждые пять обновлений дискриминатора.

Результаты тестирования алгоритмов с вероятностями ошибок первого и второго рода

приведены ниже в табл. 1. Примеры сгенерированных изображений представлены на рис. 5. Результаты, представленные в табл. 1 показывают, что точность распознавания сгенерированных АП при использовании статистического алгоритма генерации оказалась чуть выше по сравнению с алгоритмом на основе GAN сети. В целом результат для обоих алгоритмов оказался приблизительно одинаковым вне зависимости от варианта формирования обучающей выборки, то есть вне зависимости от процентного присутствия синтезированных АП, вероятности первого и второго рода очень схожи.

ЗАКЛЮЧЕНИЕ

В статье рассмотрена задача генерации АП с использованием различного рода алгоритмов. В процессе исследования был проведен сравнительный анализ двух самых универсальных и независимых от данных методов. Результатами тестирования являются показатели ошибок первого и второго рода, опи-

Таблица 1. Вероятности ошибок первого и второго рода при распознавании сгенерированных помех

[Table 1. Probabilities of errors of the first and second type when recognizing generated noise]

	$P_{err}^{(1)}$	$P_{err}^{(2)}$
Тестовая выборка, первого типа 90/10 (СА)	0.09	0.07
Тестовая выборка, второго типа 10/90 (СА)	0.07	0.06
Тестовая выборка, первого типа 90/10 (GAN)	0.11	0.08
Тестовая выборка второго типа 10/90 (GAN)	0.08	0.07

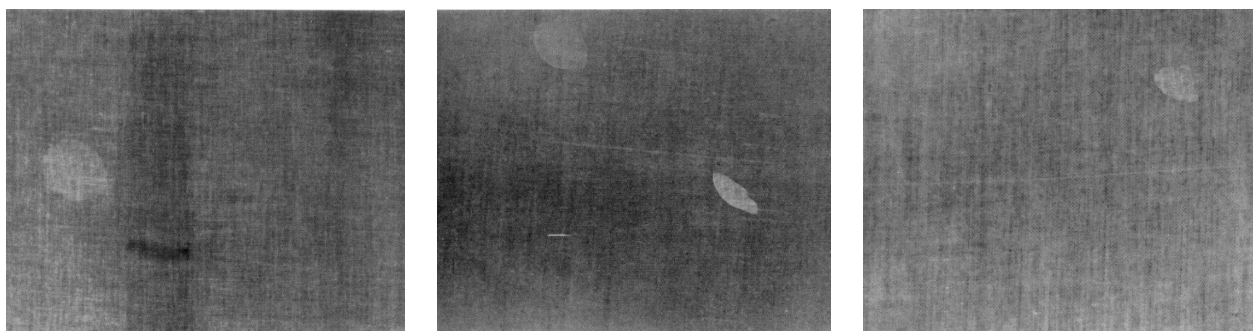


Рис. 5. Пример исходного и сгенерированных изображений
[Fig. 5 Example of the source and generated images]

раясь на эти показатели, можно сказать, что исследуемые методы генерации АП показали хороший результат, так как вне зависимости от процентного содержания синтезированных изображений НС показывала приблизительно одинаковые результаты. Так же следует отметить, что метод генерации с использованием НС класса GAN показал несколько менее успешные результаты по сравнению с алгоритмом генерации АП как локальных областей закрытия исходных данных. Однако его преимуществом является отсутствие необходимости подбора параметров синтезированных локальных образований. Исходя из полученных результатов, можно сделать вывод, что использование двух данных подходов для генерации АП можно применять для решения задач аугментации данных.

КОНФЛИКТ ИНТЕРЕСОВ

Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

СПИСОК ЛИТЕРАТУРЫ

1. Гонсалес, Р. Цифровая обработка изображений. / Р. Гонсалес, Р. Вудс. – Техносфера, 2005. – 1072 с
2. *Jeremy, S. De Bonet.* Multiresolution sampling procedure for analysis and synthesis of texture images / Jeremy S. De Bonet // Proceedings of the 24th annual conference on Computer graphics and interactive techniques – 1997 – P. 361–368.
3. *Efros, Alexei A.* Texture Synthesis by Non-Parametric Sampling / Alexei A. Efros, Thomas K. Leung // In Proceedings of the International Conference on Computer Vision-Volume 2. – 1999. – Volume 2 (ICCV '99), Vol. 2. IEEE Computer Society, Washington, DC, USA, 1033-.
4. *Калинин, П. В.* Моделирование аппликативных искажений с различной степенью прозрачности и случайной формой / П. В. Калинин, А. А. Сирота // Цифровая обработка сигналов. – 2013. – № 1. – С. 28–33.
5. *Perlin, K.* An image synthesizer / K. Perlin – SIGGRAPH. ACM. – 1985.
6. *Gatys, L.* Texture synthesis using convolutional neural networks / L. Gatys, A. S. Ecker, M. Bethge // Advances in Neural Information Processing Systems. – 2015. – P. 262–270.
7. *Zhang, G.* Defect-GAN: High-Fidelity Defect Synthesis for Automated Defect Inspection / G. Zhang [et al.] // Proceedings of the IEEE/CVF Winter Conference on Applications of Computer. – 2021.
8. *Park, T.* Gaugan: semantic image synthesis with spatially adaptive normalization / T. Park M.-Y. Liu, T.-C. Wang, J.-Y. Zhu // In ACM SIGGRAPH 2019 Real-Time Live. – 2019. – С. 1.
9. *Brock, A.* Large scale gan training for high fidelity natural image synthesis / A. Brock, J. Donahue, K. Simonyan // In ICLR. – 2018.
10. *Jun-Yan Zhu.* Unpaired image-to-image translation using cycleconsistent adversarial networks / Jun-Yan Zhu In [et al.] // ICCV – 2017. – P. 2223–2232.
11. *Yunjey, C.* StarGAN: Unified generative adversarial networks for multi-domain image / C. Yunjey [et al] // Computer Vision and Pattern Recognition. – 2018. DOI: 10.1109/CVPR.2018.00916
12. *Heeger, D. J.* Pyramid-based texture analysis/synthesis. / D. J. Heeger, J. R. Bergen // In Proceedings of the 22nd annual conference on Computer graphics and interactive techniques. – 1995. – ACM, P. 229–238.
13. *Portilla, J.* A parametric texture model based on joint statistics of complex wavelet coefficients / J. Portilla, E. P. Simoncelli // International journal of computer vision 40. – 2000. – P. 49–70.
14. *Efros, A. A.* Image quilting for texture synthesis and transfer. / A. A. Efros W. T. Freeman // In Proceedings of the 28th annual conference on Computer graphics and interactive techniques. – 2001. – ACM. P. 341–346.
15. *Liu, Y.* Near-regular texture analysis and manipulation. / Liu Y., Lin W. C., J. Hays // In ACM Transactions on Graphics (TOG). – 2004. – Vol. 23. – ACM, P. 368–376.
16. *Li, C.* Precomputed real-time texture synthesis with markovian generative adversarial networks / C. Li, M. Wand. // In Proceedings of the

14th European Conference on Computer Vision (ECCV). – 2016. – P. 702–716.

17. Liu, M. Y. Coupled generative adversarial networks / M. Y. Liu, O. Tuzel. // In Advances in Neural Information Processing Systems (NIPS). – 2016. – P. 469–477.

18. Kingma, D. P. Auto-encoding variational bayes / D. P. Kingma, M. Welling. // In Proceedings of the 2nd International Conference on Learning Representations (ICLR). – 2014.

Сирота Александр Анатольевич — д-р техн. наук, проф., зав. кафедрой технологий обработки и защиты информации Воронежского государственного университета.

E-mail: sir@cs.vsu.ru

ORCID iD: <https://orcid.org/0000-0002-5785-8513>

Дрюченко Михаил Анатольевич — канд. техн. наук, доцент кафедры технологий обработки и защиты информации Воронежского государственного университета.

E-mail: m_dryuchenko@mail.ru

ORCID iD: <https://orcid.org/0000-0001-8837-5875>

Пузатых Михаил Сергеевич — аспирант 3-го года обучения по программе «Системный анализ, управление и обработка информации (по отраслям)» факультета компьютерных наук Воронежского государственного университета.

E-mail: puzatyh.mihail@yandex.ru

ORCID iD: <https://orcid.org/0000-0003-4598-4982>

DOI: <https://doi.org/10.17308/sait/1995-5499/2022/4/87-98>

ISSN 1995-5499

Received 06.10.2022

Accepted 05.12.2022

MODELING OF APPLICATIVE NOISE IN IMAGES USING DEEP NEURAL NETWORKS

© 2022 A. A. Sirota, M.A. Dryuchenko, M.S. Puzatykh

Voronezh State University

1, Universitetskaya Square, 394018 Voronezh, Russian Federation

Annotation. The process of modeling specific defects in digital images is of great practical importance in data augmentation for trainable algorithms for image recognition, classification and analysis. Applicative noise is one of the common types of digital image distortions in technical vision systems. The effect of applicative noise is manifested in the replacement of individual sections of a useful image with fragments with random amplitude, texture and shape.

The article deals with the problem of modeling applicative noise on images using modern algorithms for statistical processing and machine learning. The main types of algorithms and methods for applicative noise modeling are briefly described. The description of the nonparametric methods for texture synthesis, methods of applicative noise generation as local occlusion areas of source images, procedures for texture synthesis using convolutional neural networks, and generative adversarial neural networks for applicative noise synthesis is given. A comparative analysis of the statistical algorithm for generating applicative noise as local areas of occlusion

with a trainable algorithm for generating images, distorted by applicative noise, based on the use of GAN models was carried out. The TILDA Textile

✉ Puzatykh Mikhail S.
e-mail: puzatyh.mihail@yandex.ru

Texture Database data set was used to test the algorithms. A method based on the use of deep convolutional classifiers for an objective assessment of the realism of the resulting applicative errors is given. The results of the study of the identity of generated and natural applicative noises in the image are analyzed.

Keywords: applicative noise, noise generation, neural networks, convolutional networks, GAN networks.

CONFLICT OF INTEREST

The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

REFERENCES

1. Gonzalez R. and Woods R. (2005) Digital Image Processing. *Tekhnosfera*. P. 1072.
2. Jeremy S. De Bonet. (1997). Multiresolution sampling procedure for analysis and synthesis of texture images. *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. P. 361–368.
3. Efros Alexei A. and Leung Thomas K. (1999) Texture Synthesis by Non-Parametric Sampling. *In Proceedings of the International Conference on Computer Vision-Volume 2 – Volume 2 (ICCV '99)*. Vol. 2. IEEE Computer Society, Washington, DC, USA, 1033-.
4. Kalinin P. V. and Sirota A. A. (2013) Modeling of applicative noise with random shape and various degrees of opacity. *Digital signal processing*. No 1. P. 28–33.
5. Perlin K. (1985) An image synthesizer. *SIGGRAPH*. ACM.
6. Gatys L., Ecker A. S. and Bethge, M. (2015) Texture synthesis using convolutional neural networks. *In Advances in Neural Information Processing Systems*. P. 262–270.
7. Zhang G, Cui K, Hung T. Y. and Lu S. (2021) Defect-GAN: High-Fidelity Defect Synthesis for Automated Defect Inspection. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer*.
8. Park T., Liu M.-Y., Wang T.-C. and Zhu J. Y. (2019) GauGAN: semantic image synthesis with spatially adaptive normalization. *In ACM SIGGRAPH 2019 Real-Time Liv* P. 1.
9. Brock A., Donahue J. and Simonyan K. (2018) Large scale gan training for high fidelity natural image synthesis. *In ICLR*.
10. Jun-Yan Zhu, Taesung Park, Phillip Isola and Alexei A Efros (2017) Unpaired image-to-image translation using cycleconsistent adversarial networks. *In ICCV*. P. 2223–2232.
11. Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo (2018) StarGAN: Unified generative adversarial networks for multi-domain image. *Computer Vision and Pattern Recognition*. DOI: 10.1109/CVPR.2018.00916
12. Heeger D. J. and Bergen J. R. (1995) Pyramid-based texture analysis/synthesis. *In Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM. P. 229–238.
13. Portilla J. and Simoncelli E. P. (2000) A parametric texture model based on joint statistics of complex wavelet coefficients. *International journal of computer vision* 40. 1. P. 49–70.
14. Efros A. A. and Freeman W. T. (2001) Image quilting for texture synthesis and transfer. *In Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM. P. 341–346.
15. Liu Y., Lin W.-C. and Hays J. (2004) Near-regular texture analysis and manipulation. *In ACM Transactions on Graphics (TOG)*. Vol. 23. ACM. P. 368–376.
16. Li C. and Wand M. (2016) Precomputed real-time texture synthesis with markovian generative adversarial networks. *In Proceedings of the 14th European Conference on Computer Vision (ECCV)*. P. 702–716.
17. Liu M.-Y. and Tuzel O. (2016) Coupled generative adversarial networks. *In Advances in Neural Information Processing Systems (NIPS)*. P. 469–477.
18. Kingma D. P. and Welling M. (2014) Auto-encoding variational bayes. *In Proceedings of the 2nd International Conference on Learning Representations (ICLR)*.

A. A. Сирота, М. А. Дрюченко, М. С. Пузатых

Sirota Alexander A. — Dr. tech. Sciences, prof., Head of the Department of Information Processing and Security Technologies at Voronezh State University.

E-mail: sir@cs.vsu.ru

ORCID iD: <https://orcid.org/0000-0002-5785-8513>

Dryuchecenko Mikhail A. — Cand. Sc. (Technology), docent at the Department of Information Processing and Security Technologies at Voronezh State University,

E-mail: m_dryuchenko@mail.ru

ORCID iD: <https://orcid.org/0000-0001-8837-5875>

Puzatykh Mikhail S. — post-graduate student, Department of Information Processing and Security Technologies at Voronezh State University

E-mail: puzatyh.mihail@yandex.ru

ORCID iD: <https://orcid.org/0000-0003-4598-4982>