

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

*Факультет Санкт-Петербургская школа физико-математических и
компьютерных наук*

Гусев Андрей Сергеевич

**ВОССТАНОВЛЕНИЕ УЧАСТКОВ СЕЙСМОГРАММ С ПОМОЩЬЮ ГЛУБОКОГО
ОБУЧЕНИЯ**

Выпускная квалификационная работа
по направлению подготовки 01.04.02 Прикладная математика и информатика
образовательная программа «Программирование и анализ данных»

Рецензент
А.И. Севостьянов

Научный руководитель
канд. техн. наук, доц.
Т.А. Брыксин
Консультанты
А.А. Шпильман
Р.А. Суханов

Санкт-Петербург 2021

Сейсмическая разведка, основанная на прохождении упругих волн через слои породы с различными свойствами, широко применяется в поиске полезных ископаемых. По различным причинам часть данных может быть повреждена и непригодна для интерпретации или просто отсутствовать. Восстановление информации о таких областях может повысить точность сейсмического исследования. Существующие решения подавляют шум, распределённый по исследованной области и не делающий информацию полностью непригодной. Другое ограничение сейсмической разведки - высокая стоимость, ограничивающая объём исследуемой почвы. В данной работе исследуется возможность восстановления отсутствующих фрагментов готовых сейсмических кубов с помощью методов дополнения изображения (inpainting), основанных на глубоком обучении, разработанных в области компьютерного зрения. Помимо заполнения пропусков в сейсмическом кубе реализовано предсказание характеристик породы в областях, близких к границе исследованной области с помощью методов расширения изображений. Обнаружено, что методы, успешно применяемые для повышения реалистичности генерируемого изображения, не повышают точность восстановления сейсмической информации. Кроме этого, реализован метод автоматической сегментации повреждённых областей на срезах сейсмических кубов.

Seismic tomography based on elastic waves passing through layers of ground with different properties, is widely used in mineral resources exploration. Due to various reasons, parts of the data may be lacking or corrupted and unsuitable for further interpretation. Restoring information about such zones could make seismic

research more accurate. Existing solutions attenuate noise that is distributed across the researched area and does not make the information unreadable. Another limitation of seismic tomography is the high cost that puts limits on the volumes of researched ground. This thesis researches the possibility to restore absent parts of finished seismic cubes using deep learning methods designed for the image inpainting task in the field of computer vision. Apart from filling missed regions in seismic cube, prediction of layers near the border of the researched area by utilizing image outpainting methods was implemented. It was discovered that methods successfully applied for improving the realism of generated images do not improve the quality of seismic information recovery. Apart from that, a model for automatic segmentation of corrupted zones was trained.

Оглавление

Введение	7
1. Анализ предметной области	10
2. Обзор литературы	11
2.1. Image inpainting	11
2.2. Image outpainting	15
2.3. Восстановление сейсмических данных	17
3. Методы	22
3.1. Глубокое обучение	22
3.1.1. Свёрточные нейронные сети	22
3.1.2. Оптимизация	23
3.1.3. Batch normalization	25
3.1.4. Dropout	26
3.1.5. PReLU	28
3.2. GAN	29
3.2.1. Wasserstein GAN	30
3.3. U-Net	31
3.4. Модули	32
3.4.1. Residual ссылки	32
3.4.2. Depthwise separable convolution	34
3.4.3. Squeeze excitation block	35
3.5. Функции потерь и метрики	35
3.5.1. Коэффициент корреляции Пирсона	35
3.5.2. Perception loss	36
3.5.3. Style loss	37

3.5.4. Dice loss	37
4. Подготовка данных	39
4.1. Inpainting	40
4.2. Экстраполяция сейсмических срезов	42
4.3. Восстановление трасс	43
4.4. Сегментация шума	45
5. Эксперименты	47
5.1. Inpainting	47
5.1.1. Классическое решение	47
5.1.2. Модели	48
5.1.3. Обучение	50
5.1.4. Результаты	50
5.2. Экстраполяция сейсмических срезов	55
5.2.1. Классическое решение	55
5.2.2. Модель	55
5.2.3. Обучение	56
5.2.4. Результаты	56
5.3. Шум в трассах	57
5.3.1. Модель и обучение	57
5.3.2. Результаты	59
5.4. Сегментация шума	60
5.4.1. Модель	60
5.4.2. Обучение	60
5.4.3. Результаты	61
Заключение	62

Голоссарий	64
Список литературы	65

Введение

Сейсморазведка – основной площадной геофизический метод по выделению перспективных нефтяных и газовых участков, а также решения ряда других геологических задач. Метод строится на физике распространения упругих волн в нижнем полупространстве. Возбуждение упругого импульса производится специальными установками или взрывом (скважинным или наземным), часть отражается от тех или иных пластов, другая часть теряет энергию и проходит дальше. Фиксация отраженных и преломленных волн осуществляется приёмниками, расположенными на земной поверхности. Полевой этап в России обычно проходит в зимнее время, т.к. данное время позволяет быть более мобильным (реки и озёра становятся более проходимыми), а также наносить меньше повреждений окружающей среде. Полевых сезонов может быть несколько, если площадь, на которой необходимо провести сейсморазведку, большая. После проведения полевых работ следует этап обработки полученных материалов, вслед за ним процесс интерпретации. В данной работе ведётся исследование по материалу, который поступил на этап интерпретации, после этапа обработки полевых материалов. Такой материал может быть как в 2D формате, так и в 3D. По мере развития обрабатываемых мощностей всё больше начинают работать именно с трёхмерными данными.

Данные после этапа обработки не всегда являются достаточно качественными, на них могут присутствовать различного рода артефакты как технического характера, так и геологического. Примером технического артефакта может быть проявление

системы разработки на глубоких пластах, как на амплитудном факторе, так и иногда на структурном. Такого рода артефакты были довольно распространены до того, как появились более продуктивные алгоритмы и более мощное аппаратное обеспечение, тем не менее такого рода артефакты до сих пор можно встретить на сейсмических данных, оставшихся с тех времён. Другой тип артефактов связан с геологическим характером среды, где, например, наблюдаются необъяснимые объекты, связанные с поведением границ или наличием дифрагированных волн. Для повышения качества дальнейшей интерпретации тот или иной артефакт было бы полезно удалить. С этим процессом связано первое направление данной работы - восстановление удалённого участка данных внутри имеющегося сейсмического объёма.

Стоимость проведения такого рода работ оценивается в несколько сотен миллионов рублей, включая полевые работы, обработку данных и интерпретацию. Процесс обычно занимает 2-3 года. Проведение таких работ планируется заранее с наиболее эффективной системой полевой расстановки, положением приёмников и источников, однако, в некоторых местах проведение такого рода работ запрещено, либо процесс согласования занимает длительный период, например, любого рода заказники или заповедники, населённые пункты, объекты стратегического назначения (аэропорты, железнодорожные вокзалы и т. п.). При этом, имея максимально полный объём данных о геологии региона, можно значительно снизить неопределённости при построении геологических моделей и дальнейшего бурения. С этим связано второе направление данной работы – восстановление неизвестной

части сейсмических данных за пределами имеющегося объёма.

В области компьютерного зрения существуют направления image inpainting и image outpainting, в рамках которых решаются задачи восстановления отсутствующих областей в изображениях и предсказания по доступному изображению содержания за его пределами. Значительные результаты в повышении реалистичности восстанавливаемой части изображения достигнуты с помощью генеративно-состязательных нейронных сетей, а также специальных слоёв и функций потерь.

В текущей работе дополнение пропусков в сейсмических срезах различной формы реализовано с помощью методов глубокого обучения, разработанных для решения задачи image inpainting. Реализована экстраполяция сейсмической информации за пределы доступного среза как решение задачи image outpainting. Обнаружено, что применение состязательного обучения и дополнительных функций потерь не повышает точность предсказания сейсмической информации. Также в данной работе реализована сегментация повреждённых участков сейсмических срезов, которые могут требовать дополнения.

1. Анализ предметной области

Скорость распространения упругой волны зависит от свойств горной породы, включая состав, плотность и др.. По значениям этой скорости породы можно разбить на относительно однородные слои. Границы таких слоёв зачастую совпадают с геологическими границами. На границе сред с различной скоростью волны и плотностью часть волны отражается, а часть - преломляется. Отражённая волна проходит ряд преломляющих границ и регистрируется приёмниками. При обработке зарегистрированных приёмниками сигналов восстанавливаются различия в скоростях прохождения волн через различные породы, а также определяется переход от времени фиксации сигнала к глубинам отражающих границ. Формируется сейсмический куб, представляющий характеристику пропускающей способности породы в зависимости от расположения и глубины. Такие данные позволяют определять однородные объёмы, которые затем помогают определять расположение пород с теми или иными свойствами.

Некоторые объекты могут закрывать среду в некоторой окрестности, препятствуя прохождению упругих волн - полностью поглощая или полностью отражая их. В результате информация об отдельных областях в сейсмическом кубе полностью или частично отсутствует. Восстановление такой отсутствующей информации - одна из целей данной работы.

Поскольку применяемые методы были разработаны для обработки изображений, в данной работе сейсмические кубы рассматриваются как наборы двумерных срезов.

2. Обзор литературы

2.1. Image inpainting

Ряд работ [15, 5, 17] применяет генеративно-сопоставительные сверточные модели для дополнения изображений. В этих работах генератор получает на вход изображение с удалённым фрагментом, и обучается восстанавливать исходное изображение, обманывая генератор, который одновременно обучается различать восстановленные и исходные изображения. Функция потерь состоит из сопоставительного компонента, и компонента, ответственного за точность восстановления, который представляет собой L1 или L2 расстояние между исходным и восстановленным изображением.

Некоторые работы разделяют дискриминатор на две части, из которых одна отвечает за глобальные свойства восстановленного изображения, а другая - за локальные. Так, в статье Иизуки и соавторов [15] дискриминатор разделяется на два: один принимает изображение полностью, другой - только восстановленную часть или соответствующую часть оригинального изображения. На определённом слое активации дискриминаторов конкатенируются и передаются в финальный классификатор, который определяет, оригинальное получено изображение или восстановленное.

В работе Демира и Унала [5], напротив, есть два дискриминатора, по выходам которых по отдельности вычисляются сопоставительные функции потерь генератора L_{g_adv} и L_{p_adv} :

$L_{generator} = \lambda_{rec}L_{rec} + \lambda_{g_adv}L_{g_adv} + \lambda_{p_adv}L_{p_adv}$. Первый дискри-

минатор отвечает за глобальные признаки и принимает полное изображение. Второй отвечает за локальные признаки. Он работает как дискриминатор PatchGAN с ограниченной областью восприятия. При этом дискриминаторы разделяют первые слои для более эффективного извлечения низкоуровневых признаков. Архитектура генератора в данной статье включает свёртки с шагом, превышающим единицу для понижения разрешения, residual блоки, не изменяющие размерность, и свёртки с интерполяцией для восстановления разрешения.

Статья Патхак и соавторов [2] вводит контекстный кодировщик. Эта архитектура развивает традиционный автокодировщик: кодировщик принимает на вход изображение с пропущенной областью и вычисляет скрытое представление, из которого декодировщик должен восстановить исходное изображение. Кодировщик повторяет архитектуру свёрточных слоёв AlexNet, хотя и не следует распространённой практике использования предобученных на задаче классификации на датасете ImageNet весов, а обучается с нуля. Нововведение данной статьи - поканальный полносвязный слой между кодировщиком и декодировщиком. Он позволяет в явном виде передать информацию о контексте с сохранённой части изображения к удалённой. Полносвязный слой применяется поканально для оптимизации объёма модели и использования вычислительных ресурсов: если обозначить число каналов как m , а размер карты признаков как $n \times n$, то количество весов для поканального слоя составляет mn^4 , в то время как для простого полностью связанного слоя - m^2n^4 .

Другой подход предлагается в [32]. Эта работа вводит меха-

низм *shift connection*. Основой модели является сеть U-Net. На одной из размерностей промежуточных карт признаков применяется перенос данных между слоем кодировщика Φ_l и соответствующим слоем декодировщика Φ_{L-l} . Для каждой позиции y в карте признаков на выходе Φ_l , относящейся к удалённой части, выбирается позиция x из сохранённой части, такая что косинусное сходство соответствующих им векторов признаков $\Phi_{L-l}(y)$ и $\Phi_l(x)$ максимально. Затем формируется карта признаков Φ_{L-l}^{shift} из соответствующих удалённым позициям сохранённых векторов с Φ_l : $\Phi_{L-l}^{shift}(y) = \Phi_l(x)$. Она конкатенируется с Φ_{L-l} и передаётся в следующий блок. Другая особенность данной работы - функция потерь *guidance loss*. Авторы обращают внимание на то, что для позиций y , относящихся к сохранённой части изображения, выполняется приблизительное равенство $\Phi_{L-l}(y) \approx \Phi_l(y)$. *Guidance loss* стремится обеспечить похожее условие для позиций в картах признаков, относящихся к удалённой части изображения. Функция *guidance loss* равняется квадрату расстояния L2 между сужениями карт признаков Φ_{L-l} и Φ_l на удалённую часть изображения Ω : $L_{guidance} = \sum_{y \in \Omega} \|\Phi_l(y) - \Phi_{L-l}(y)\|_2^2$.

Среди классических методов дополнения изображений, применявшихся до распространения глубокого обучения, часто применялось копирование небольших фрагментов сохранённой части изображения, которые признаются наиболее вероятными в удалённой части. Сочетанием такого подхода с современным глубоким обучением является статья Янга и соавторов [12]. В ней заполнение пропуска разбивается на два этапа: высокоуровневое предсказание содержания и уточнение текстуры. Пред-

сказание содержания выполняемое с помощью генеративно-состязательной модели, похожей на те, которые были реализованы в описанных выше работах. Уточнение текстуры производится с помощью дополнительной "текстурной" нейросети (использовалась предобученная VGG-19). Восстанавливаемая часть разбивалась на компактные фрагменты размером 3×3 . Для каждого из фрагментов находился наиболее близкий по активациям заданных слоёв текстурной нейросети в сохранённой части изображения, после чего расстояние между ними минимизировалось. Оптимизация выполнялась итеративно с чередованием этапов при помощи алгоритма BFGS.

Назери и соавторы предлагают свой подход к дополнению изображения с разделением на два этапа в [7]. Сначала достраиваются границы изображения. Это делается с помощью генеративно-состязательной модели, генератор которой принимает на вход чёрно-белую версию изображения маску и карту границ сохраняемой части. Затем оригинальное изображение восстанавливается другой генеративно-состязательной моделью, принимающей изображение с пропуском и предсказанную карту границ.

Статья Лиу и соавторов [16] фокусируется на восстановлении регионов неправильной формы. Она вводит частичную свёртку - блок, который в явном виде поддерживает маску, пикселей, значение которых определено. На входе сети в маске единицы стоят в пикселях, которые не были удалены. На каждом последующем слое единицы в маске выставляются для точек, в которых скользящее окно свёртки включает хотя бы один определённый пиксель входной карты признаков. Значение x' выхо-

да частичной свёртки можно определить для точек, в которых скользящее окно свёртки включает хотя бы один определённый пиксель, как $x' = W^T(X \odot M) \frac{sum(1)}{sum(M)} + b$, где: W - веса свёртки, X - значения входных пикселей, попадающие в скользящее окно для x' , M - маска определённых пикселей, для скользящего окна свёртки, b - смещение свёртки, \odot - операция поточечного умножения. В остальных точках значение свёртки равно нулю. Функция потерь в данной работе имеет три компонента помимо расстояния до целевого изображения: perception loss, style loss и total variance loss.

В статье Ю и соавторов [8] предлагается gated convolution, блок, похожий на partial convolution, но не передающий маску в явном виде, а определяющий её автоматически. В нём, помимо основной свёртки, задаётся дополнительная со значениями в диапазоне от 0 до 1 (сигмоида в качестве функции активации). Выходом блока является произведение результатов основной и дополнительной свёрток. Формально gated convolution можно описать так: $O(x) = \phi(F(x)) \odot \sigma(G(x))$, где O - результат gated convolution, ϕ - функция активации основной свёртки, F - основная свёртка, σ - функция активации дополнительной свёртки, сигмоида, G - дополнительная свёртка.

2.2. Image outpainting

Сабини и Русак в [29] применяют генеративно-состязательные сети для предсказания содержания изображения за пределами известной области. В данной работе самая левая и самая правая четверти изображения удаляются, и освободившееся место за-

полняется средним значением оставшейся части изображения. Генератор принимает на вход изображение с удалёнными краями и маску. В качестве генератора используется полностью свёрточная сеть, включающая одну свёртку с понижением размерности и одну обратную операцию, восстанавливающую размерность карты признаков. Дискриминатор в данной работе имеет два пути: глобальный, принимающий изображение полностью, и локальный, получающий левую половину изображения или горизонтальное отражение правой части изображения. После некоторого количества слоёв активации с разных путей конкатенируются.

Подход к экстраполяции изображений с помощью генеративно-состязательных сетей развивается Сяо и соавторами в [36]. Авторы данной работы используют в качестве генератора U-Net с deep residual block. Deep residual block представляет собой объединение идей residual соединений и dense блоков. Как и в densenet, вход каждого слоя связан с выходами всех предыдущих слоёв. Сходство с resnet заключается в способе соединения - выходы разных слоёв не конкатенируются, как в densenet, а складываются.

Идея копирования фрагментов исходного изображения для сборки отсутствующей части находит применение в статье Ван Хорика [13]. В данной работе экстраполяция происходит в два этапа. Сначала похожим на вышеописанным способом, с помощью генеративно-состязательной модели экстраполируется содержание изображения. Для повышения качества предсказанной части изображения применяется гармонизация моделью SinGAN, обученной на исходном изображении. Модель SinGAN

обучается на одном изображении и позволяет переносить текстуры исходного изображения на другое, сохраняя общую структуру этого изображения.

Статья Кима и соавторов [22] сводит задачу экстраполяции изображения к задаче заполнения пропусков. Части изображения переупорядочиваются таким образом, что удалённая часть оказывается не снаружи, а внутри изображения. В таком виде отсутствующая часть заполняется в два этапа: сначала предсказываются границы на пропущенном участке, затем, используя предсказанные границы, восстанавливается сама отсутствующая часть. После этого расположение частей изображения восстанавливается.

2.3. Восстановление сейсмических данных

Глубокое обучение применялось для улучшения качества данных сейсмической разведки. Существует ряд работ, использующих методы подавления шума на сейсмограммах.

Ю и соавторы в [37] обучают полностью свёрточную нейронную сеть для восстановления зашумлённых сейсмических данных. В качестве цели используются сейсмограммы, качество которых признано удовлетворительным. На вход модели подаются те же самые фрагменты в искажённом виде: применяется наложение белого гауссовского шума по всей площади фрагмента, а также линейный шум - гиперболические линии, не соответствующие существующим горизонтам.

Похожим образом действуют Джинь и соавторы в [31]: нейронная сеть архитектуры автокодировщик с residual ссылками

обучается восстанавливать фрагменты сейсмограмм, на которые накладывается гауссовский шум.

Ричардсон и Феллер в [26] решают задачу разделения накладывающихся волн. Наложение волн возникает, когда в одно и то же время в одном районе работает несколько источников. Такая ситуация возникает в условиях ограниченного времени на исследования. В данной работе синтетический датасет генерируется наложением одних срезов сейсмических данных на другие. Нейронная сеть обучается предсказывать наложенный срез. Также авторы накладывают на срезы сейсмических кубов шум и показывают, что модель, обученная на наложенных срезах, способна выделять и его.

Сленг и соавторы в [34] накладывают на сейсмограммы волны от других источников, а также от других ударов данного источника и показывают применимость глубоких нейронных сетей для удаления таких пересечений.

В работе Гадильшина, Сильвестрова и Бакулина [9] алгоритмы дополнения изображений интегрируются в процесс обработки сейсмограмм. Создаётся маска сложной формы, объединяющая артефакты различного происхождения (недостатки обработки, повреждённые данные). Некоторые области удаляются для повышения производительности. По сохранённой части данных вычисляются атрибуты для следующего этапа обработки. Применяется кодирование этих атрибутов цветами изображения. С помощью нейронной сети, использующей слои *partial convolution* заполняются пропуски в изображении, которое затем декодируется обратно в параметры сигнала.

Эти методы работают с сейсмограммами, которые требуют

трудоёмкой обработки перед тем, как будет создан сейсмический куб, пригодный для дальнейшего анализа; данные сейсмограммы не всегда доступны. Кроме того, [37], [31] и [26] работают с шумом умеренной амплитуды, равномерно распределённым по сейсмограмме, что не позволяет обрабатывать случаи, когда данные по ограниченному району оказываются полностью непригодны для использования. Обработка таких случаев представляет собой задачу заполнения пропусков в изображениях, исследованную в области компьютерного зрения.

В работе Лиу и соавторов [25] разрабатывается метод очистки шумных сейсмических кубов. В качестве входа используется сейсмический куб в том виде, в котором он был получен в результате обработки, а в качестве целевого сигнала - результат обработки этого куба традиционными методами подавления шума. Авторы обосновывают такой подход тем, что реальный шум, встречающийся в сейсмических данных, отличается от белого гауссовского шума, применяемого в работах по подавлению шума методами глубокого обучения. Глубокая нейронная сеть обучается предсказывать разницу между очищенной и исходной версиями сейсмического куба. Как и в описанных выше работах, в данном случае происходит повышение качества существующих данных, но не делается предсказания об областях, информация про которые полностью отсутствует.

Жанг и соавторы предлагают ещё один способ очистки сейсмических данных с помощью глубоких нейронных сетей в [24]. Отличие их метода в том, что для его обучения не требуются чистые данные. Рассматриваются двумерные сейсмические срезы. Используемая нейронная сеть представляет собой простой ав-

токодировщик, включающий свёрточные слои, пулинговые слои и слои, восстанавливающие размерность. Автокодировщик обучается восстанавливать переданный срез, но он не обучается сохранять шум, не содержащий каких-либо закономерностей.

Ву и соавторы предлагают в [35] ещё один способ удаления шума с сейсмических кубов. Их метод также не требует чистых данных для обучения. Он состоит из четырёх этапов. Сначала методами цифровой обработки данных (variational mode decomposition) сейсмический куб переводится в альтернативное представление (intrinsic mode functions) из нескольких компонент. На втором этапе к этим компонентам добавляется синтетический белый гауссовский шум. Затем свёрточная нейронная сеть обучается выделять добавленный шум. На последнем этапе обученная модель применяется к данным до добавления синтетического шума, чтобы выделить содержащийся в них естественный шум. Предсказанный шум вычитается из данных, а результат переводится в исходное представление.

Ли и соавторы применяют восстановление изображений с помощью генеративно-состязательных нейронных сетей для сжатия сейсмических кубов в [30]. Сжатие куба достигается через удаление значительной доли сейсмических трасс (степень сжатия варьируется между 2 и 16, то есть остаётся от $\frac{1}{2}$ до $\frac{1}{16}$ сейсмических трасс исходного куба). Рассматриваются различные способы выбора подмножества трасс для сохранения: равномерный выбор с варьируемым шагом, случайный выбор, а также собственный способ оптимизации подмножества трасс для сохранения для достижения точности восстановления оригинального куба. Затем сейсмический куб восстанавливается с

помощью модели архитектуры Wasserstein GAN. Исследуется перенос обучения генератора с моделей, предназначенных для работы с естественными изображениями и показывается, что такие веса не подходят для работы с данными сейсмической разведки из-за различий в характере изображений. Такой подход отличается от целей данной работы тем, что сохраняемые точки располагаются регулярно, то есть для любого удалённого значения существует сохранённое значение на небольшом расстоянии. Это другая задача, близкая к *suprerresolution*, повышению разрешения изображения.

3. Методы

3.1. Глубокое обучение

3.1.1. Свёрточные нейронные сети

Свёрточные нейронные сети предназначены областей, в которых признаки имеют пространственное расположение, и более близкие признаки взаимодействуют более интенсивно. Если в полносвязной нейронной сети каждый нейрон связан со всеми нейронами предыдущего слоя, то в свёрточной сети нейрон зависит только от активаций из одной компактной области. Веса представляются в виде матрицы, которая перемещается по массиву входных признаков скользящим окном. Для изображений применяются двумерные свёртки. Значение нейрона на выходе двумерного свёрточного слоя описывается как $X'_{u,v,c} = \delta(\sum_{i<n, j<m, k<d_{in}} X_{u+i, v+j, k} * W_{i,j,k} + b)$, где X - входная карта признаков, X' - карта признаков на выходе, W - веса, также называемые ядром свёртки, $n \times m$ - размер ядра, b - смещение, δ - функция активации, d_{in} - число каналов на входе, d_{out} - число каналов на выходе. Ядро имеет размерность $d_{out} \times n \times m \times d_{in}$.

Часто в свёрточных нейронных сетях возникает необходимость перехода к карте признаков меньшего разрешения. Один из способов добиться этого - свёртка с шагом, отличным от 1. Шаг - это число точек, на которое смещается ядро свёртки между вычислениями соседних значений выхода. Другой способ - пулинг. При пулинге значения активаций группируются по расположению, а затем по ним вычисляется функция, которая может быть максимумом (max pool) или средним (average

pooling).

В некоторых ситуациях, например, в нейронных сетях архитектуры автокодировщик или при генерации изображений из векторов, возникает необходимость переходить к карте признаков большего разрешения. Для этого можно применять транспонированные свёртки, при которых ядро свёртки перемещается по шагам в выходной карте признаков и изменяет каждое из её выбранных значений на произведение текущего значения входной карты признаков и соответствующего элемента ядра. Также транспонированную свёртку можно представить как добавление пропусков между позициями в карте признаков и последующее применение обычной свёртки. Транспонированные свёртки склонны приводить к появлению повторяющихся с периодом, равным шагу, узоров, известных как артефакты шахматной доски. По этой причине чаще используется интерполяция карты признаков до необходимого разрешения и применение обычной операции свёртки.

3.1.2. Оптимизация

Основой для современных методов оптимизации в глубоком обучении является метод градиентного спуска. Он работает с функцией $f(w)$, оптимизируя параметры w так, чтобы минимизировать $f(w)$. Оптимизация происходит итеративно. На каждом шаге вычисляется производная f при текущих параметрах и делается шаг против производных: $w' = w - \eta f'(w)$, где η - гиперпараметр, называемый скоростью обучения. В глубоком обучении оптимизируемая функция представляет собой сум-

му функций потерь по элементам обучающей выборки: $f(w) = \sum_{x \in X} L(x, w)$. Как правило, вычисление градиентов по всем функциям потерь невозможно из-за ограничений по используемой оперативной памяти. Эту проблему решает стохастический градиентный спуск: на каждом шаге выбирается один объект обучающей выборки, вычисляется производная функции потерь в нём и выполняется обновление весов. Недостатком такого алгоритма является высокая дисперсия градиента по одному образцу. На практике применяется компромиссное решение: на каждом шаге оптимизации вычисляется градиент по пакету - подмножеству обучающей выборки небольшого размера.

В описанном виде градиентный спуск страдает от сильной склонности к застреванию в локальном минимуме. Для уменьшения этого недостатка применяется момент: обновление выполняется не напрямую в направлении против градиента в текущей точке, а в направлении скорости, определяемой как сумма момента - предыдущей скорости, умноженной на вес, и градиента в текущей точке. Формально:

$$\nu' = \nu * \mu + \nabla_w f(w)$$

$$w' = w - \eta * \nu,$$

где ν - скорость, μ - гиперпараметр, регулирующий влияние момента. Усовершенствованием метода стохастического градиентного спуска с моментом является момент Нестерова. При использовании момента Нестерова градиент вычисляется в точке, соответствующей шагу из текущей точки по текущему моменту $\nu' = \nu * \mu + f'(w - \eta\nu)$.

Другая идея по улучшению градиентного спуска - уменьшать скорость обучения для весов, по которым пройден большой путь. Это увеличивает эффективность в сценариях, где редкие признаки имеют большое значение. Формально:

$$c' = c + (\nabla_w f(w))^2$$

$$w' = w - \frac{\eta}{\sqrt{c + \epsilon}} \nabla_w f(w),$$

где ϵ - гиперпараметр сглаживания для предотвращения деления на ноль. Неограниченно возрастающая сумма квадратов градиентов может слишком сильно снижать скорость обучения. Эту проблему решает оптимизатор Adadelta, в котором сумма заменяется на экспоненциальное скользящее среднее: $c' = \gamma c + (1 - \gamma)(\nabla_w f(w))^2$, γ - гиперпараметр. Оптимизатор Adam совмещает подход Adadelta с моментом - шаг делается по моменту, а скорость обучения делится на скользящее среднее квадратов градиентов:

$$v' = v * \mu + \nabla_w f(w)$$

$$c' = \gamma c + (1 - \gamma)(\nabla_w f(w))^2$$

$$w' = w - \frac{\eta}{\sqrt{c + \epsilon}} v.$$

3.1.3. Batch normalization

Распределение входов слоя может изменяться по мере обучения в зависимости от порядка предъявления объектов обучающей выборки. Это изменение называется ковариантным сдвигом. Ковариантный сдвиг негативно влияет на процесс обучения: требуется постоянная корректировка весов под новое рас-

пределение входов. Для борьбы с ковариантным сдвигом применяется пакетная нормализация. При пакетной нормализации математическое ожидание каждой активации поддерживается равным нулю, а стандартное отклонение - единице. Для этого для каждого выхода на каждом пакете обучающей выборки вычисляется среднее и эмпирическое стандартное отклонение. Затем из выходов вычитается среднее, а результат делится на стандартное отклонение (для предотвращения деления на 0 применяется сглаживание):

$$x^{(k)} \leftarrow \frac{x^{(k)} - E(x^{(k)})}{\sqrt{D(x^{(k)}) + \epsilon}}.$$

Для проведения пакетной нормализации при выполнении нейронной сети во время обучения вычисляются средние значения математического ожидания и дисперсии. Отсюда появляется различное поведение в режимах обучения и выполнения: если при обучении для нормализации используются среднее и дисперсия пакета, а глобальные статистики обновляются, то при выполнении нормализация происходит по глобальным статистикам, а их обновление не производится. Помимо ковариантного сдвига пакетная нормализация решает проблему взрывающихся градиентов.

3.1.4. Dropout

В машинном обучении часто возникает проблема переобучения - явления, когда модель адаптируется к закономерностям, специфичным для обучающей выборки, из-за чего качество мо-

дели на генеральной совокупности снижается. Склонность к переобучению возрастает с увеличением сложности модели. Для борьбы с переобучением применяются методы регуляризации, ограничивающие сложность модели или задающие априорное распределение параметров. Наиболее распространённым способом регуляризации для нейронных сетей является dropout. Идея dropout в том, чтобы зашумить взвешенную сумму активаций в нейроне. Для этого применяется добавление шума, имеющего распределение Гаусса или Бернулли, к весам связей нейронов или к активациям (из центральной предельной теоремы следует эквивалентность этих подходов, так как значение, перед функцией активации является суммой с большим числом слагаемых и имеет нормальное распределение). Общепринятым является dropout с распределением Бернулли, применяемый к активациям. Каждая активация умножается на случайную величину, имеющую распределение Бернулли с параметром p . Таким образом, получается слой, в котором каждый выход заменяется на ноль с заданной вероятностью p . При выполнении нейронной сети, чтобы максимально эффективно использовать все обученные веса, замена активаций нулями не применяется. При этом простой отказ от отключения части активаций ведёт к смещению математического ожидания взвешенных сумм активаций в нейронах следующего слоя. Для предотвращения такого смещения при вычислении слоя dropout в режиме выполнения все активации умножаются на p .

3.1.5. PReLU

Функция активации ReLU (Rectified linear unit) равна самому входному значению, если оно положительно, и нулю для отрицательных входных значений: $ReLU(x) = \max(x, 0)$. Она получила распространение в сфере глубокого обучения из-за эффективности вычисления и отсутствия затухания градиентов при больших входных значениях. Однако, проблема исчезновения градиентов всё ещё присутствует: градиент ReLU в отрицательных значениях равен нулю. Из-за этого при обратном распространении ошибки градиент не проходит через часть нейронов, а некоторые нейроны остаются не затронуты процессом обучения. Для исправления этой проблемы предлагались различные приближения ReLU. Одно из них - LeakyReLU:

$$LeakyReLU(x) = \begin{cases} x, & \text{if } x > 0 \\ ax, & \text{if } x \leq 0 \end{cases}$$

где a - гиперпараметр. Производная LeakyReLU отлична от нуля, но эксперименты показывают, что она даёт незначительное улучшение результатов по сравнению с ReLU [4]. Дальнейшее усовершенствование ReLU - PReLU (Parametric Rectified linear unit). Формула PReLU выглядит похожим образом, но a является не гиперпараметром, а обучаемым параметром. PReLU может давать больший по модулю градиент в отрицательных числах, а может обучиться работать как ReLU (если выученный параметр a равен нулю) или LeakyReLU (при малом по модулю параметре a).

3.2. GAN

Описанные выше работы по дополнению изображений применяют фреймворк GAN [11], генеративно-состязательных сетей, в котором обучаются две нейросети: генератор, который должен создавать образцы так, чтобы их распределение как можно меньше отличалось от распределения реальных образцов, и дискриминатор, который обучается отличать настоящие образцы от сгенерированных. Для обучения генератора применяется состязательная функция потерь: генератор обучается генерировать такие примеры, чтобы дискриминатор классифицировал их как настоящие. Для этого выход генератора передается дискриминатору, для выхода дискриминатора вычисляется функция потерь как для реальных примеров. Обратным распространением ошибки через дискриминатор вычисляются градиенты для оптимизации генератора. Функцию, оптимизируемую при обучении GAN можно описать как:

$$\min_G \max_D E_{data}(\log D(x)) + E_{noise}(\log(1 - D(G(z))))$$

В классическом GAN генератор принимает вектор случайного шума и отображает распределение этого шума в целевой распределение. Существует модификация GAN - условные генеративно-состязательные модели (Conditional GAN), которые принимают дополнительную информацию y . Это может быть класс или какие-либо другие свойства объекта. Они позволяют явно контролировать создаваемые объекты. Функция

потерь имеет следующий вид:

$$\min_G \max_D E_{data}(\log D(x|y)) + E_{noise}(\log(1 - D(G(z|y))))$$

Для генерации новых изображений из заданного класса применялся DCGAN, соединяющий генеративно-сопоставительные и свёрточные нейронные сети. В нём вектор шума последовательно преобразуется свёртками и интерполяцией.

3.2.1. Wasserstein GAN

Классический GAN страдает от ряда проблем при обучении. Проблема коллапса мод заключается в том, что генератор всегда возвращает один и тот же результат вне зависимости от шума на входе. Дискриминатор может обучиться классифицировать эту моду как сгенерированную, но в таком случае генератор лишь обучается возвращать другую фиксированную моду. Другая проблема - затухание градиентов. Если дискриминатор обучается качественно разделять настоящие и сгенерированные примеры, а генератор не успевает обучиться качественной генерации, может возникнуть ситуация, когда дискриминатор возвращает для сгенерированных образцов близкие к предельным вероятности, а производная функции активации в соответствующих значениях близка к нулю, что приводит к тому, что модули градиентов, доходящих до генератора также оказываются близкими к нулю, из-за чего обучение генератора становится невозможным.

Развитием подхода GAN, призванным исправить описанные недостатки, является Wasserstein GAN [1]. Обычный GAN при-

ближает распределение генерируемых данных к распределению реальных по мере Йенсона-Шенона. Wasserstein GAN обучается более стабильно за счёт того, что приближает реальное распределение по расстоянию Вассерштайна. Это достигается через обеспечение 1-липшицевости дискриминатора, который для Wasserstein GAN принято называть критиком. Для ограничения константы Липшица существует несколько методов. Первый - обрезка весов. При использовании данного метода на модули весов нейросети задаются ограничения, и веса, модули которых превышают данное значение, обрезаются. К недостаткам такого подхода можно отнести возможное затруднение обучения. Развитием Wasserstein GAN, обеспечивающим более стабильное обучение, является gradient penalty [18]. При нём к функции потерь критика добавляется штраф за разницу между градиентом и единицей. Также существует спектральная нормализация [33] - деление матрицы весов на операторную норму.

3.3. U-Net

Для задач сегментации изображений часто применяется модель U-Net, предложенная в работе Ронненбергера и соавторов по сегментации изображений, полученных в результате медицинских исследований [27]. U-Net - это полностью свёрточная нейронная сеть, в которой разрешение карты признаков сначала снижается, а затем восстанавливается. Схематическое изображение архитектуры представлено на рис. 1 Снижающая и восстанавливающая разрешение части являются последовательно блоками одинаковой длины. После каждого блока снижа-

ющей разрешение части применяется операция максимального пулинга, снижающая разрешение карты признаков в четыре раза. Перед каждым блоком восстанавливающей разрешение части применяется операция интерполяции, увеличивающая разрешение карты признаков. Между снижающей и повышающей разрешение частями располагается дополнительный блок, называемый бутылочным горлышком, который не изменяет размерность карты признаков. Между блоками первой и последней частей с совпадающими размерностями выходной и входной карты признаков существуют ссылки: выход блока перед пулингом соединяется с выходом предыдущего блока или бутылочного горлышка после восстановления разрешения и передаётся на вход следующему блоку. Такая архитектура позволяет, с одной стороны, сохранить способность автокодировщика обрабатывать контекст, а с другой - сохранить детали, которые иначе были бы потеряны при операции пулинга.

3.4. Модули

3.4.1. Residual ссылки

Архитектура resnet, предложенная в [3], улучшает сходимость глубоких нейронных сетей. Она решает проблему затухания градиентов при распространении через множество слоёв с помощью соединений между выходами различных слоёв. Оригинальный residual блок можно описать как $H(x) = \delta(F(x) + x)$, где $H(x)$ - выход residual блока. $F(x)$ - два свёрточных слоя с функцией активации между ними, δ - функция активации. Схема такого блока представлена на рис. 2. В случае, если необхо-

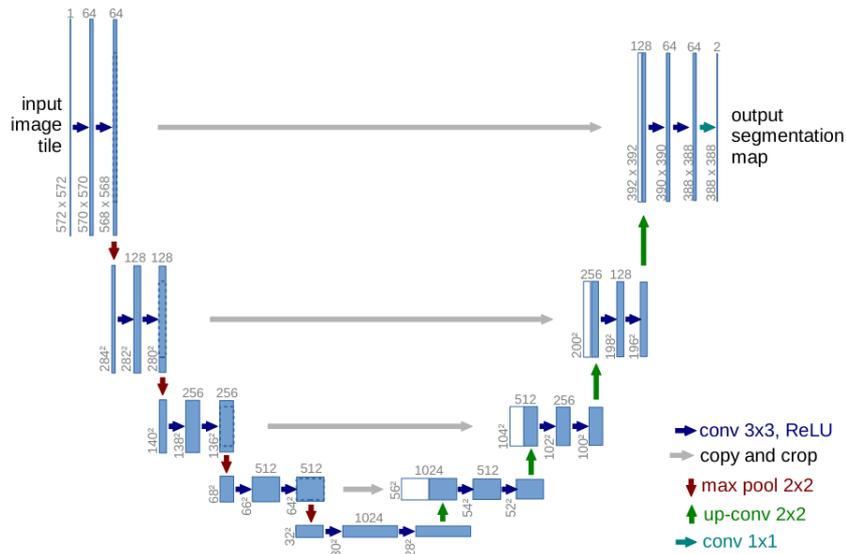


Рис. 1: Архитектура U-Net. Источник - статья U-Net: Convolutional Networks for Biomedical Image Segmentation [27]

димо обеспечить изменение числа каналов в карте признаков, число каналов на выходе блока должно отличаться от числа каналов на входе, добавляется изменение размерности ссылки: ко входной карте признаков применяется свёртка с ядром размерности 1×1 с необходимым количеством каналов на выходе. Тогда блок имеет вид $H(x) = \delta(F(x) + D(x))$, где $D(x)$ - свёрточный слой с ядром 1 и необходимой размерностью выходной карты признаков. Такие ссылки уменьшают число преобразований, которые должен пройти градиент, чтобы попасть на первые слои.

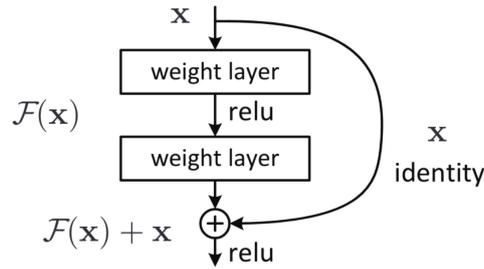


Рис. 2: Архитектура residual блока. Источник - статья Deep Residual Learning for Image Recognition [3]

3.4.2. Depthwise separable convolution

Блок Depthwise Separable convolution был предложен в [20] для сокращения числа параметров свёртки и вычислительных операций при её вычислении. Он состоит из двух шагов: сначала свёртка с размером ядра больше 1 применяется отдельно к каждому каналу. Такая свёртка позволяет обработать пространственное расположение факторов. Затем свёртка с ядром 1×1 применяется ко всем каналам. Эта свёртка позволяет учесть взаимодействие различных факторов в одном пространственном расположении. Число параметров для depthwise separable convolution составляет $kernel_size * input_channels + input_channels * output_channels$, где $kernel_size$ - размерность ядра, а $input_channels$ и $output_channels$ - количество каналов во входной и выходной картах признаков, в то время как для обычной свёртки это число составляет $kernel_size * input_channels * output_channels$.

3.4.3. Squeeze excitation block

Блоки Squeeze & excitation были предложены в [14] и применены в задаче сегментации в [28]. Они усиливают более полезные и одновременно заглушают менее полезные признаки, умножая карту признаков на автоматически определяемые веса. Существует три типа блоков:

- Блок channels squeeze & excitation вычисляет веса каналов и полностью умножает на них соответствующие каналы карт признаков. Веса каналов вычисляются двуслойной полносвязной нейронной сетью $\bar{z} = W_2(\delta(W_1(z)))$ по вектору средних активаций по каналам: $z_k = \frac{1}{HW} \sum_i \sum_j u_k(i, j)$.
- Блок spatial squeeze & excitation оценивает значение каждого пикселя свёрткой с ядром размерности 1×1 и умножает на него соответствующие значения в карте признаков для всех каналов.
- Блок spatial and channel squeeze & excitation вычисляет как веса каналов, так и веса пикселей, и умножает каждое значение в карте признаков на произведение веса канала и веса расположения.

3.5. Функции потерь и метрики

3.5.1. Коэффициент корреляции Пирсона

Коэффициент корреляции Пирсона является показателем зависимости двух величин. Он изменяется в диапазоне от -1 до

1. Значение 1 соответствует линейно зависимым величинам с положительной зависимостью (с ростом одной из величин увеличивается и другая), значение -1 означает отрицательную линейную зависимость, 0 соответствует независимым величинам. Коэффициент корреляции Пирсона двух случайных величин ξ и ν определяется как $\rho_{\xi,\nu} = \frac{cov(\xi,\nu)}{\sigma_\xi\sigma_\nu}$, где $cov(\xi,\nu)$ - ковариация, а σ_ξ и σ_ν - стандартные отклонения величин. В данной работе коэффициент корреляции используется в качестве метрики качества.

3.5.2. Perception loss

Функция perception loss, введённая в [10], представляет собой сумму расстояний между активациями дополнительной нейросети (в статье используется модель VGG16, обученная на датасете ImageNet) для исходного и восстановленного изображения: $L_{perceptual} = \sum_{p \in P} \frac{\|\Psi_p^{out} - \Psi_p^{gt}\|}{N} + \frac{\|\Psi_p^{comp} - \Psi_p^{gt}\|}{N}$, где P - множество используемых слоёв вспомогательной нейросети, Ψ_p^{out} - активации p -го слоя вспомогательной сети на выходе модели, Ψ_p^{comp} - активации p -го слоя вспомогательной сети на заполненном изображении (заполненное изображение - это исходное изображение, в котором удалённая часть заменена соответствующей частью предсказания), Ψ_p^{in} - активации p -го слоя вспомогательной сети на исходном изображении до удаления части пикселей. По сравнению с прямым вычислением расстояния между изображениями, такая функция потерь позволяет учитывать базовую семантику изображения.

3.5.3. Style loss

Функция потерь Style loss - это нормализованное расстояние между нормированными матрицами Грама активаций вспомогательной сети. Матрицей Грама системы векторов $V = v_1, \dots, v_n$ называется матрица, состоящая из скалярных произведений векторов этой системы: $G_{i,j} = \langle v_i, v_j \rangle$.

$$L_{style} = \sum_{p \in P} \frac{1}{C_p^2} \|K_p(\Psi_p^{outT} \Psi_p^{out} - \Psi_p^{gtT} \Psi_p^{gt})\| + \\ + \sum_{p \in P} \frac{1}{C_p^2} \|K_p(\Psi_p^{compT} \Psi_p^{comp} - \Psi_p^{gtT} \Psi_p^{gt})\|,$$

где $K_p = \frac{1}{C_p H_p W_p}$ - нормализационная константа.

3.5.4. Dice loss

В задачах сегментации изображений функцией потерь часто является попиксельная кросс-энтропия. Её недостатком является чувствительность к сбалансированности данных: если один класс значительно превосходит другие по распространению, модель, предсказания которой смещены в сторону этого класса. Для оценки сегментации можно применять метрику F1, но она не является дифференцируемой и, следовательно, не может выступать в качестве функции потерь. Обычно проблема несбалансированности данных решается заданием различных весов для классов вручную или функцией от частоты встречаемости: более редкий класс получает больший вес. Другой способ решения этой проблемы предлагает Dice loss, предложенная в [19] - функция потерь для задач сегментации, приближающая метрику

ку F1 (Dice coefficient) непрерывной величиной. Она определяется как

$$\frac{2 \sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2}$$

, где p_i - предсказание модели для пикселя i , а g_i - целевое значение для этого пикселя.

4. Подготовка данных

Для создания датасетов для обучения и оценки моделей использовалось несколько источников сейсмических срезов. Были использованы чистые сейсмические кубы F3 и Penobscot, а также изображения срезов из различных источников.

Сейсмический куб F3 опубликован в [21] как часть открытого датасета для задачи сейсмической интерпретации. Куб представляет собой результат разведки в Северном море на расстоянии 180 километров от побережья Дании. Он имеет разрешение 580 на 951 на 462 отсчётов. Расстояние между отсчётами составляет 25 метров, таким образом, куб содержит информацию о породе на глубине до 11550 метров на площади 14500 на 23775 метров. Для генерации синтетического датасета для задаче сегментации использовалась его уменьшенная версия разрешением 226 на 251 на 176 отсчётов.

Сейсмический куб Penobscot также опубликован как часть открытого датасета для сейсмической интерпретации [23]. Он получен на Шотландском Шельфе и имеет разрешение 460 на 481 на 800 отсчётов. С учётом расстояния между отсчётами в 25 метров, куб содержит информацию о породе на площади до 11500 на 12025 метров и на глубине до 20000 метров.

Помимо сейсмических кубов использовался набор из 33 изображений срезов, собранных из различных источников. Несмотря на низкое количество изображений, некоторые из них давали большое количество примеров при нарезке на фрагменты за счёт высокого разрешения. Так, самое большое из изображений имело разрешение 6944 на 4232 пикселя, что давало 432 квадра-

та со стороной 256. В отличие от указанных выше сейсмических кубов эти изображения содержали зашумлённые области, которые были размечены для создания датасета для задачи сегментации, а также использовались для получения форм областей для удаления при генерации датасета для дополнения неправильной формы.

4.1. Inpainting

Для задачи inpainting использовался сейсмический куб F3. Поскольку используемая модель работает с двумерными данными, ей передавались срезы куба. Датасеты готовились двумя способами.

Первый способ - удаление областей неправильной формы. Из куба F3 брались как инлайны, срезы по первой координате, так и кросслайны, срезы по второй координате. Чтобы избежать повторения, использовались срезы с шагом 20. Помимо срезов куба, использовались изображения. Срезы и изображения нарезались на квадраты со стороной, равной степени двойки, но не меньше 256. Все фрагменты масштабировались до разрешения 256 на 256 пикселей. Такое многоуровневое разделение применялось с целью получить изображения с различной частотой слоёв. Из изображений 22 были использованы для тренировочной выборки, 4 - для валидационной и 3 - для тестовой. Срезы разделялись на выборки по координате в соотношении 60 процентов в тренировочную выборку, 20 - в валидационную и 20 - в тестовую. Таким образом, получилось: 1391 образец в тренировочной выборке, 58 образцов в валидационной выборке и 620

образцов - в тестовой выборке. Среди фрагментов были вручную отобраны примеры с качественным сигналом. После этого осталось 888, 46 и 378 примеров в тренировочной, валидационной и тестовой выборках. Удалялось два типа областей: прямоугольные области и области в форме участков шума, отмеченных при разметке для сегментации. В качестве прямоугольных областей использовались прямоугольник с соотношением сторон 1 к 2, повернутый вертикально или горизонтально, а также квадрат. Для каждого фрагмента генерировалась новая форма: выбирался прямоугольник или форма, выделенной шумной области. Затем изменялось соотношение сторон: высота области умножалась на величину 2^x , $x \sim U[-1, 1]$. После этого области масштабировались так, чтобы занимать от 5 до 10 процентов площади фрагмента. Для форм областей, размеченных как шум, также применялся поворот на случайный угол. Затем применялись аугментации: горизонтальное отражение с вероятностью 0.5 и аффинное преобразование с поворотом на случайный угол от -60 до 60 градусов и масштабированием в 0.8-1.5 раз. Для каждого изображения генерировалось 10 аугментированных версий.

Второй способ - удаление областей правильной формы. В этом случае использовались только инлайны сейсмического куба F3. Удалённая область всегда занимала одну и ту же часть входа модели. Для предсказания в тестовой выборке удалялась центральная часть каждого инлайна, составляющая 5 процентов его площади. Размер вырезанной части составлял 103×213 отсчётов. На вход модели подавались фрагменты в виде квадрата со стороной 256, в центре которых находилась вырезан-

ная часть. Тренировочная и валидационная выборки формировались из квадратных фрагментов срезов, не пересекающихся с фрагментами, использованными в тестовой выборке. В этих фрагментах, как и в тестовой выборке, удалялась центральная часть размером 103×213 отсчётов. Таким образом, удалённая часть занимала приблизительно 0.33 от площади фрагмента. К тренировочной выборке применялись аугментации: горизонтальное отражение с вероятностью $\frac{1}{2}$ и поворот на случайный угол из равномерного распределения от -30 до 30 градусов. Для каждого исходного примера генерировалось 4 примера с аугментациями. Поскольку расположение удалённой части на изображении не должно было изменяться, аугментации применялись до удаления. Были опробованы также масштабирование и изменение перспективы, однако они не показали улучшения результатов.

В обоих случаях перед передачей в модель изображения нормализовались так, чтобы среднее значение сохранённой части равнялось нулю, а стандартное отклонение - единице.

4.2. Экстраполяция сейсмических срезов

В задаче *outpainting* требуется, имея некоторое изображение, достроить его, то есть распространить информацию за его границы. В данной работе такая постановка используется для получения представления о строении почвы за пределами исследованной области.

Для создания датасета *outpainting* были использованы инлайны сейсмического куба Penobscot. В тестовом датасете для

предсказания были выбраны приблизительно 5 процентов в левой части каждого инлайна и 5 процентов с правой (по 24 трассы с каждой стороны). Модель принимала на вход фрагменты срезов в форме квадрата со стороной 192. Для предсказания удалялась самая левая часть изображения шириной в 24 пикселя, и ставилась задача её восстановления. Фрагменты тестового датасета из правой части разреза горизонтально отражались так, что удалённая часть оказывалась слева. При необходимости дополнения куба правая часть горизонтально отражалась, передавалась в модель, а после восстановления снова отражалась. Таким образом, данные для предсказания всегда занимали одну и ту же часть образца. Как и в предыдущем случае, изображения были нормализованы по среднему и стандартному отклонению сохранённой части.

Обучающая и валидационная выборки формировались из частей инлайнов, не используемых в тестовой выборке для предсказания, но пересекались с частями фрагментов тестовой выборки, которые не удалялись и передавались модели на вход. Каждый фрагмент использовался как напрямую, так и в горизонтально отражённом виде для увеличения объёма выборки.

4.3. Восстановление трасс

Первым подходом к восстановлению повреждённого сигнала было восстановление узких полос из одной или нескольких трасс, пропущенных или искажённых шумом.

Для создания датасета использовались как сейсмический куб F3, использованный для генерации датасета для задач заполне-

ния пропусков правильной и неправильной формы, так и изображения сейсмических срезов. Искажениям подвергались отдельные трассы или группы до 5 трасс. Изначально трассы выбирались с постоянным шагом. Затем применялось смещение, концентрирующее трассы с левой или правой стороны фрагмента. Смещение было реализовано через перевод координат, выбранных для искажения трасс, в доли от ширины фрагмента (номер трассы делился на ширину образца) и возведение в случайную степень от 0.8 до 1.25. Применяемые искажения включали замену сигнала нулями, добавление белого шума, добавление шума Перлина и смещение данных вверх на 1-6 пикселей. Применялось искажение не по всей трассе, а на 10-80% высоты. Пример такого изображения - на рис. 3. Количество трасс подбиралось так, чтобы они совокупно занимали приблизительно 33 процента площади среза. Срезы нарезались на квадраты со стороной 128, затем применялись аугментации: горизонтальное отражение с вероятностью 0.5, случайное изменение перспективы, случайное аффинное преобразование с поворотом на угол до 20 градусов в случайную сторону и масштабированием с коэффициентом от 0.8 до 1.25.

Был реализован и более простой подход: удаление трасс с определённым шагом. Поскольку вертикальное расположение трасс не позволяет использовать такие аугментации, как поворот и искажение перспективы, применялись только горизонтальное отражение и вертикальное масштабирование. Готовые изображения проходили нормализацию по среднему и стандартному отклонению.

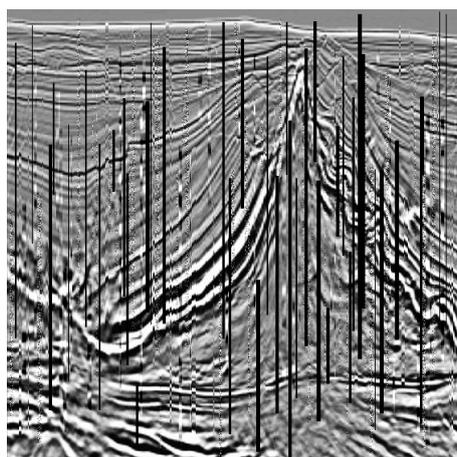


Рис. 3: Пример изображения с искажёнными трассами.

4.4. Сегментация шума

В задаче сегментации повреждённых участков использовались два подхода к получению данных: генерация синтетического датасета и разметка реальных примеров.

Для генерации синтетических данных использовался заведомо чистый сейсмический куб, уменьшенная версия F3. На его срезы накладывался синтетический шум, при этом сохранялась маска этого шума. Шум накладывался в различных формах: прямоугольники, треугольники в верхней части среза и колонки. Шум семплировался из различных распределений: белый гауссовский шум, шум Перлина, шум из фиксированных значений. Использовались как инлайны, так и кросслайны. Синтетические изображения нормализовались вычитанием среднего и делением на стандартное отклонение чистого среза.

В качестве реальных данных использовались изображения сейсмических срезов. Изображения размечались вручную. Та-

кой подход имеет ограничения: субъективность при разметке, связанная с различной природой шума: можно по-разному определять проблемные участки, требующие восстановления. Помимо участков с шумным сигналом из типичной структуры слоёв выделяются области компактного залегания пород специфического состава. Для того чтобы снизить влияние этой субъективности, разметка выполнялась двумя людьми, а затем значения вероятностей пикселей были отнесёнными к повреждённой части усреднялись.

Разметка осуществлялась помощью утилиты VGG Image Annotator [6]. VGG Image Annotator - это свободно распространяемая (под лицензией BSD 2 clause) утилита, предназначенная для разметки изображений, видеозаписей и аудиозаписей. Вокруг областей с шумным сигналом отмечались многоугольники. Для дальнейшей обработки они переводились в маски с помощью библиотеки `opencv`.

Для создания примеров, непосредственно передаваемых в модель для её обучения изображения нарезались на квадраты со стороной 256. За счёт высокого разрешения некоторых изображений это увеличивало количество примеров до 5814. Для увеличения объёма данных применялись аугментации: горизонтальное отражение с вероятностью 0.5, случайное изменение перспективы, случайное аффинное преобразование с поворотом на угол до 20 градусов в случайную сторону и масштабированием с коэффициентом от 0.8 до 1.25. На каждый фрагмент генерировалось 20 аугментированных вариантов. Все изображения нормализовались через вычитание среднего и деление на стандартное отклонение.

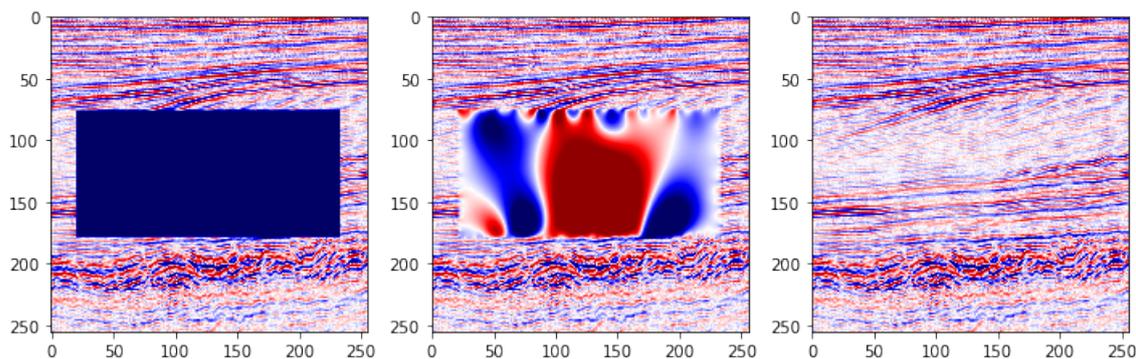


Рис. 4: Пример работы классической интерполяции с областью правильной формы. Слева находится изображение фрагмента с удалённым участком, посередине - результат бигармонической интерполяции, справа - исходный фрагмент

5. Эксперименты

5.1. Inpainting

5.1.1. Классическое решение

Для сравнения был рассмотрен классический метод интерполяции изображений, не требующий обучения. Использовался метод бигармонических функций, реализованный в библиотеке `scikit-image`. Пример его работы для областей неправильной формы представлен на рис. 5, для областей правильной формы - рис. 4. Коэффициент корреляции на тестовой выборке для неправильной формы составил 0.53. Для правильной формы - менее 0.01.

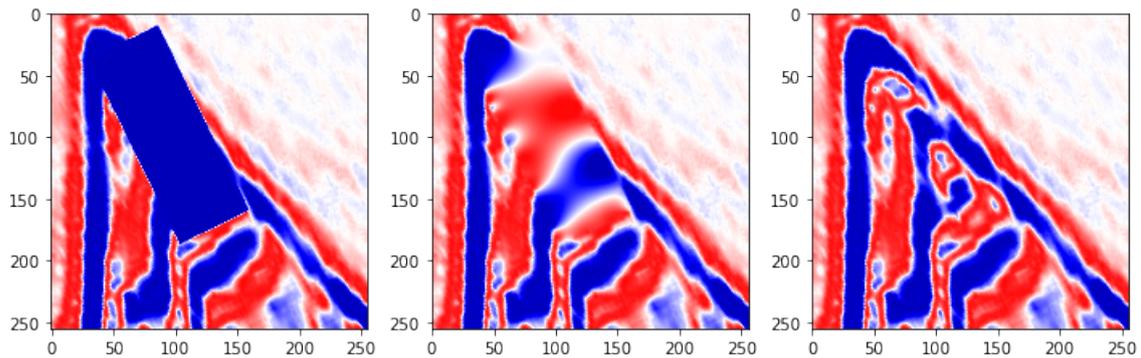


Рис. 5: Пример работы классической интерполяции с областью неправильной формы. Слева находится изображение фрагмента с удалённым участком, посередине - результат бигармонической интерполяции, справа - исходный фрагмент

5.1.2. Модели

Для дополнения участков неправильной формы применялась модель U-Net. Она была выбрана из-за необходимости в точной передаче границ, которую могут обеспечить прямые ссылки с начальных слоёв высокого разрешения на поздние слои. Для адаптации сети к отсутствию данных по части изображения в блоках генератора использовались слои gated convolution, специально разработанные для задачи image inpainting. Применялось как простое обучение с учителем, так и обучение в качестве генератора GAN, как это делается в большинстве работ по теме image inpainting. Для предотвращения коллапса мод применялся подход Wasserstein GAN с gradient penalty для обеспечения необходимой константы Липшица для критика. Для модели, обучаемой в качестве генератора WGAN рассматривались дополнительные функции потерь perception loss и style loss. Perception loss и style loss вычислялись через активации крити-

ка на трёх свёрточных слоях, поскольку типичные предобученные сети, используемые в работах по дополнению изображений, предназначаются для работы с естественными изображениями, а не с сейсмическими разрезами. Такой вариант perception loss упоминается в литературе под названием feature map loss. Общая функция потерь генератора модели с дополнительными функциями потерь имела вид $L_{gen} = \lambda_{adv}L_{adv} + \lambda_{rec}L_{rec}$, где $\lambda_{adv} = 0.1$ - вес состязательных потерь, а $\lambda_{rec} = 1.0$ - вес потерь MAE.

Во всех моделях использовалась функция потерь реконструкции MAE. В литературе по генерации изображений проводилось сравнение мер L1 (MAE) и L2 (монотонно зависит от MSE) в качестве функции потерь реконструкции, и было показано, что применение меры L1 делает изображения менее смазанными. Это можно объяснить тем, что L2 быстрее возрастает по мере удаления предсказания модели от целевого значения, подталкивая модель к "безопасным" значениям, близким к центру распределения. По этой причине в современных работах, как правило, используется именно L1 мера.

Для участков правильной формы была сделана попытка применения генеративно-состязательной модели, но результаты оказались неудовлетворительными. В результате использовалась модель U-Net. В ней применялись различные блоки: с residual ссылками, а также слоями gated convolution и partial convolution, также разработанных для задачи image inpainting. В качестве функции потерь использовалась MAE. Для предотвращения переобучения применялся слой dropout с вероятностью 0.3. На всех скрытых слоях функцией потерь была PReLU.

5.1.3. Обучение

Модели для дополнения областей неправильной формы обучались оптимизатором Adam. Применялись различные скорости обучения для разных частей: $1e-4$ - для генератора и $2e-3$ - для критика (для генеративно-сопоставительной модели).

Модели для областей правильной формы обучались оптимизатором Adam в течение 500 эпох.

Обучение происходило на видеокарте Nvidia Quadro P6000 с 24 гигабайтами видеопамяти. 500 эпох обучения модели для областей правильной формы занимали приблизительно пять часов.

Модель со слоями Partial convolution и residual ссылками, показавшая наилучший результат, обучалась повторно в течение 900 эпох. Её результаты использованы в примерах.

5.1.4. Результаты

Для областей неправильной формы метрики представлены в таблице 1. Для оценки на тестовой выборке использовалась модель с лучшей эпохи. Лучшей эпохой считалась эпоха с наименьшим значением функции потерь реконструкции на валидационной выборке. Примеры работы различных моделей представлены на рис. 6. Изображения, дополненные моделью Wasserstein GAN, выглядят более чётко, однако, простой U-Net показал более высокие метрики. Модель, обученная с использованием дополнительных функций потерь perception loss и style loss показала худшие результаты по сравнению с моделями, обученными более простыми способами.

Таблица 1: Метрики предсказания области неправильной формы внутри изображения

Модель	MAE	Коэффициент корреляции
Обычная	0.48	0.69
WGAN	0.51	0.66
WGAN + perception/style loss	0.66	0.47

Для удаления областей правильной формы метрики качества на валидационной выборке не улучшались монотонно по мере обучения. Сначала они улучшались, затем в некоторых случаях снижались, а затем выходили на плато (график коэффициента корреляции на рис. 8). При этом визуально изображения предсказаний модели продолжали становиться более правдоподобными. По этой причине указывается два показателя: на лучшей эпохе и на последней эпохе. Лучшая эпоха выбиралась по MAE на валидационной выборке. На рис. 7 представлены примеры предсказаний на примерах из тестовой выборки с последней и лучшей эпох.

Наилучшие результаты показала модель без критика и дополнительных функций потерь, но со слоями Partial convolution и residual ссылками. Метрики различных моделей представлены в таблице 2.

На основании модели для восстановления области правильной формы было реализовано заполнение пропуска в кубе через последовательное дополнение всех необходимых срезов.

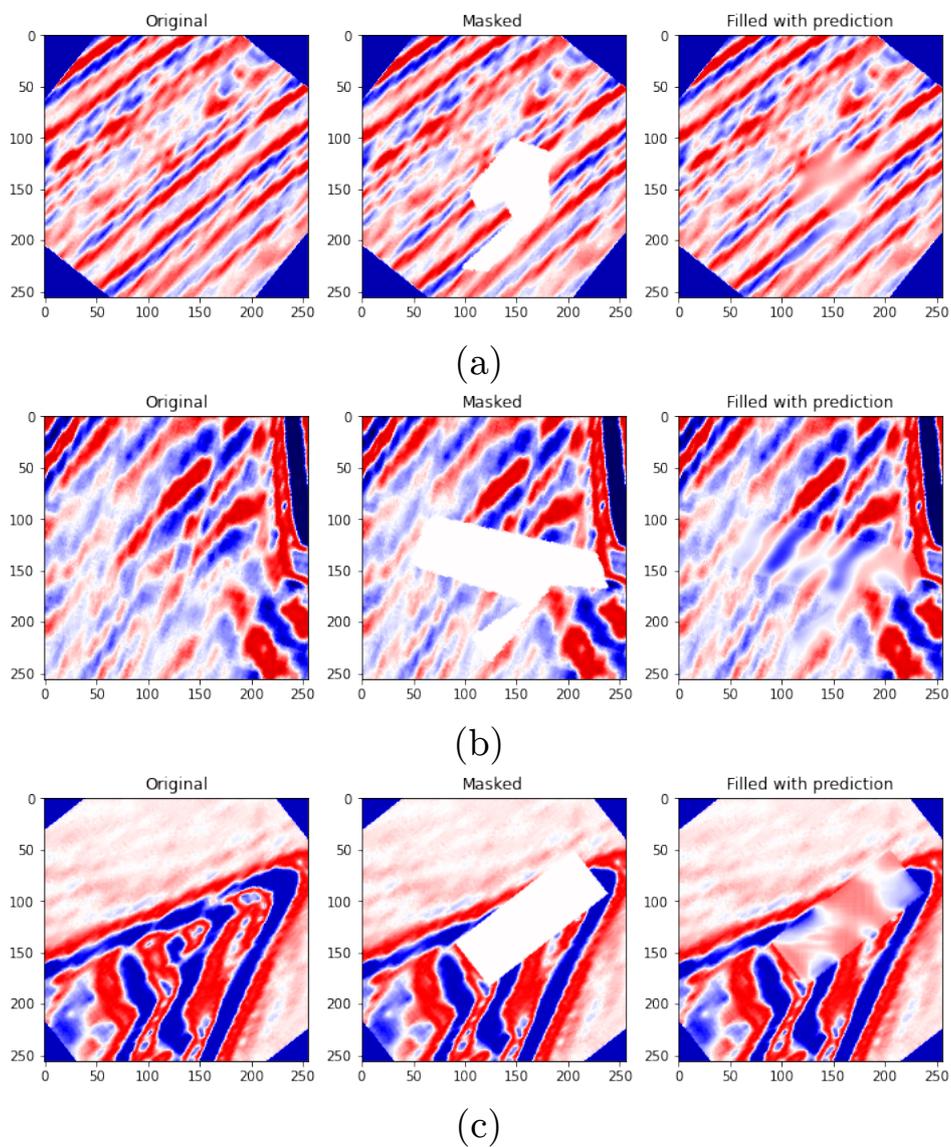
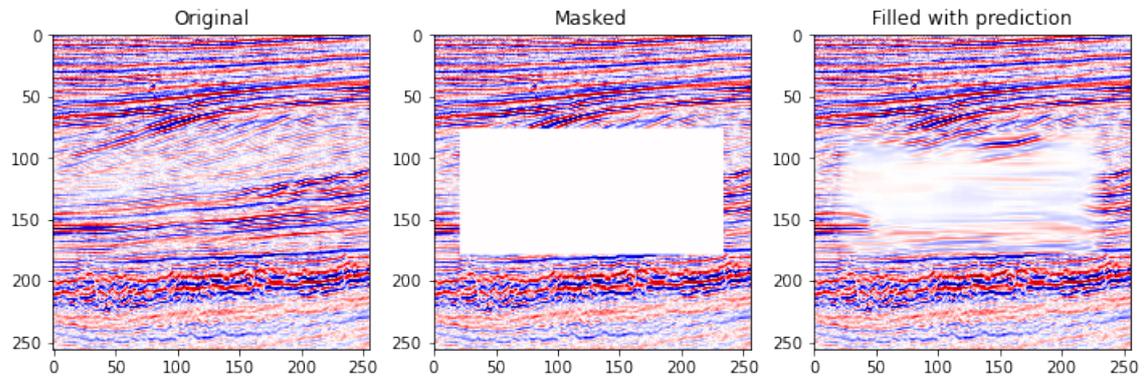
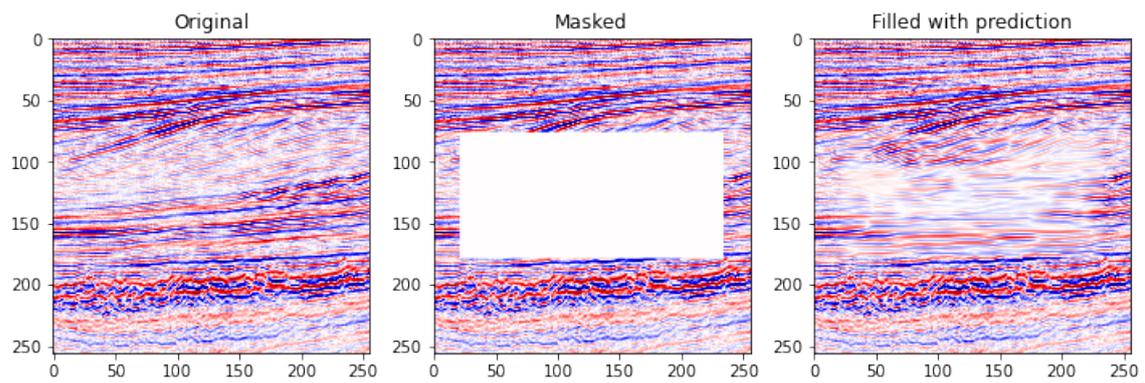


Рис. 6: Примеры работы inpainting с областями неправильной формы. (a) - простой U-Net со слоями gated convolution, (b) - модель, обученная как генератор Wasserstein GAN, (c) - модель, обученная как генератор Wasserstein GAN с дополнительными функциями потерь perception loss и style loss



(a)



(b)

Рис. 7: Примеры работы inpainting с областями правильной формы, модель со слоями Partial convolution и residual ссылками. (a) - результат модели с лучшей эпохи, (b) - результат модели с последней эпохи

Таблица 2: Коэффициент корреляции предсказания области правильной формы внутри изображения

Модель	Тестовый	Валидационный
Простой U-Net, лучшая	0.18	0.36
Простой U-Net, последняя	0.18	0.36
Residual, лучшая	0.21	0.38
Residual, последняя	0.16	0.33
Partial convolution, лучшая	0.19	0.38
Partial convolution, последняя	0.14	0.37
Partial convolution, residual, лучшая	0.22	0.39
Partial convolution, residual, последняя	0.16	0.37
Gated convolution, лучшая	0.14	0.38
Gated convolution, последняя	0.17	0.36

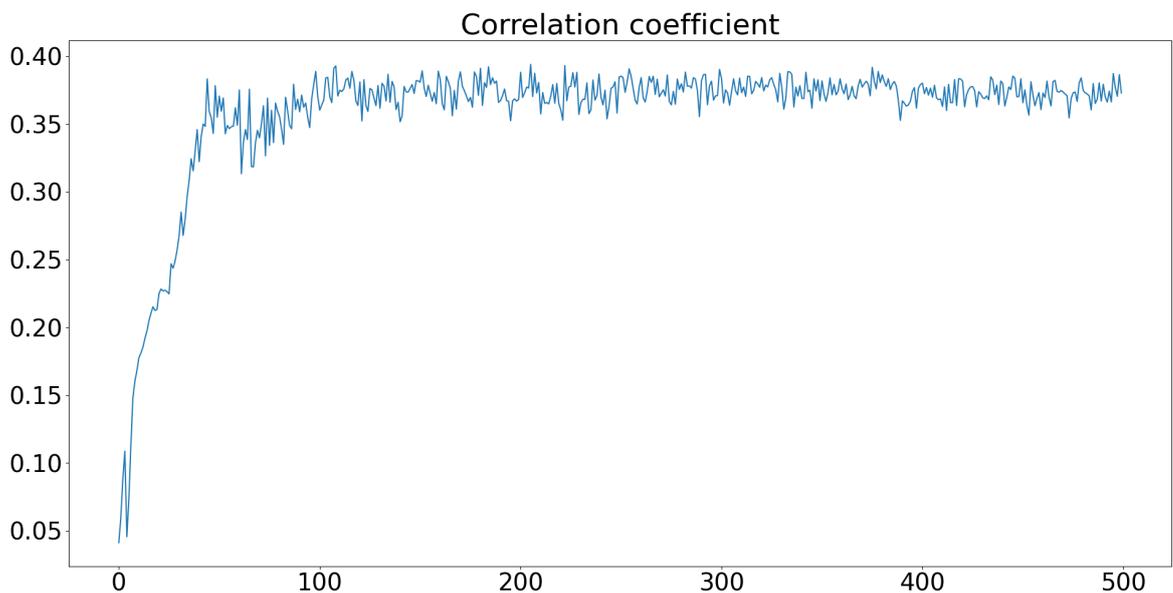


Рис. 8: График коэффициента корреляции для задачи inpainting на валидационной выборке

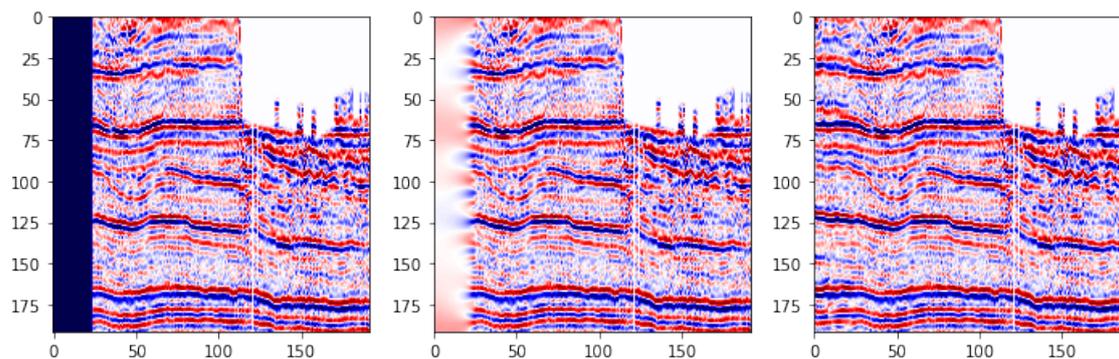


Рис. 9: Пример работы классической интерполяции для задачи экстраполяции. Слева находится изображение фрагмента с удалённым участком, посередине - результат бигармонической интерполяции, справа - исходный фрагмент

5.2. Экстраполяция сейсмических срезов

5.2.1. Классическое решение

Для сравнения был также, как и в случае задачи заполнения пропуска, рассмотрен не требующий обучения метод бигармонических функций, реализованный в библиотеке `scikit-image`. Он показал неудовлетворительные результаты: пример работы представлен на рис. 9. Коэффициент корреляции на тестовой выборке составил 0.37.

5.2.2. Модель

Как и в предыдущем случае, применялась архитектура U-Net. Модель обучалась в роли генератора Wasserstein GAN. Для обеспечения необходимой константы Липшица критика применялся метод `gradient penalty`. В качестве модуля использовался блок из двух слоёв `gated convolution`. Для уменьшения переобу-

чения перед gated convolution применялся слой dropout с вероятностью 0.2.

Помимо состязательной функции потерь, для генератора использовались функция потерь восстановления изображения - MAE по удалённой части, а также style loss и perception loss, вычисляемые по активациям критика. Вес состязательных потерь был равен 0.5, perceptual loss и style loss - 0.1 и 120 соответственно. В результате функция потерь имела вид $MAE + 0.5L_{adv} + 0.1L_{perceptual} + 120L_{style}$.

Помимо полной модели с критиком и дополнительными функциями потерь обучались модели без perception loss и style loss, а также простая модель без применения состязательного обучения.

5.2.3. Обучение

Модель обучалась в течение 200 эпох оптимизатором Adam. Скорость обучения была установлена в $1e-2$ для критика и $3e-4$ для генератора. Обучение на видеокарте Nvidia Quadro P6000 занимало приблизительно час.

5.2.4. Результаты

Как и в случае с восстановлением пропусков правильной формы, начиная с некоторого момента при обучении метрики перестают расти, но предсказанные части срезов продолжают становиться более реалистичными за счёт увеличения контрастности. При использовании состязательного обучения такой эффект можно объяснить: критик продолжает совершенствоваться

Таблица 3: Коэффициент корреляции предсказания экстраполяции за пределы среза

Модель	Тестовая	Валидационная
Лучшая, WGAN + losses	0.53	0.59
Последняя, WGAN + losses	0.52	0.56
Лучшая, WGAN	0.52	0.60
Последняя, WGAN	0.53	0.55
Лучшая, простой U-Net	0.53	0.59
Последняя, простой U-Net	0.53	0.53

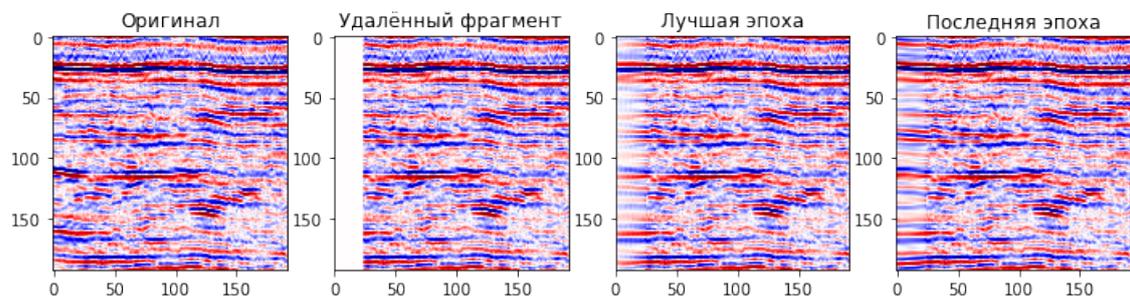
ся и подталкивать генератор к более реалистичным предсказаниям после того, как достигнута максимальная точность восстановления. Коэффициент корреляции Пирсона на тестовом и валидационном датасетах представлен в таблице 3. Из результатов видно, что дополнительные функции потерь и составительное обучение не оказали значимого влияния на метрики на тестовой выборке. Примеры работы различных моделей для экстраполяции среза представлены на рис. 10. Видно, что предсказания, полученные от более простых моделей, визуально больше отличаются от настоящих данных.

На основании полученной модели было реализовано предсказание окрестности куба, пример работы - на рис. 11.

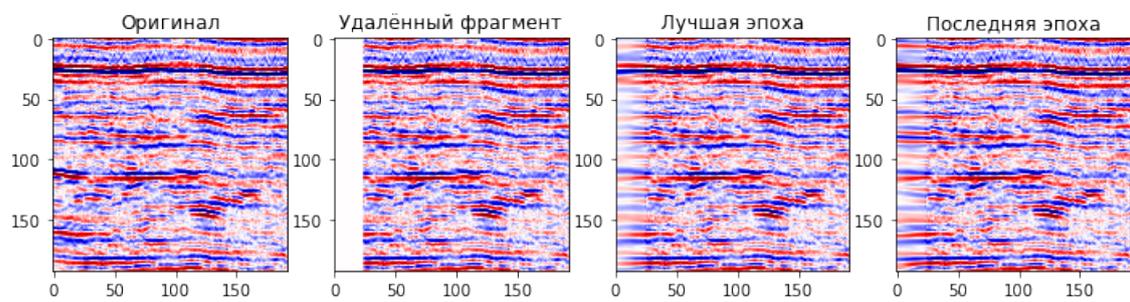
5.3. Шум в трассах

5.3.1. Модель и обучение

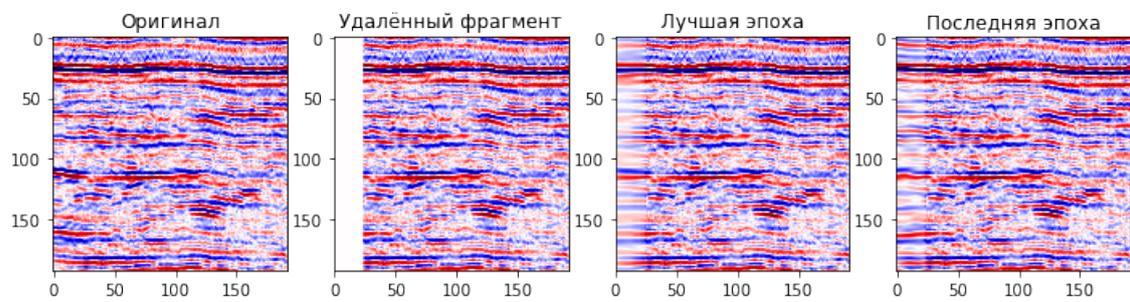
Использовалась модель U-Net с residual ссылками внутри блоков. В качестве функции потерь оптимизировалась MSE по искажённой части фрагмента. Обучение происходило в течение



(a)



(b)



(c)

Рис. 10: Примеры работы экстраполяции среза. (a) - модель, обученная как генератор Wasserstein GAN с дополнительными функциями потерь perception loss и style loss, (b) - модель, обученная как генератор Wasserstein GAN без perception loss и style loss, (c) - простой U-Net со слоями gated convolution

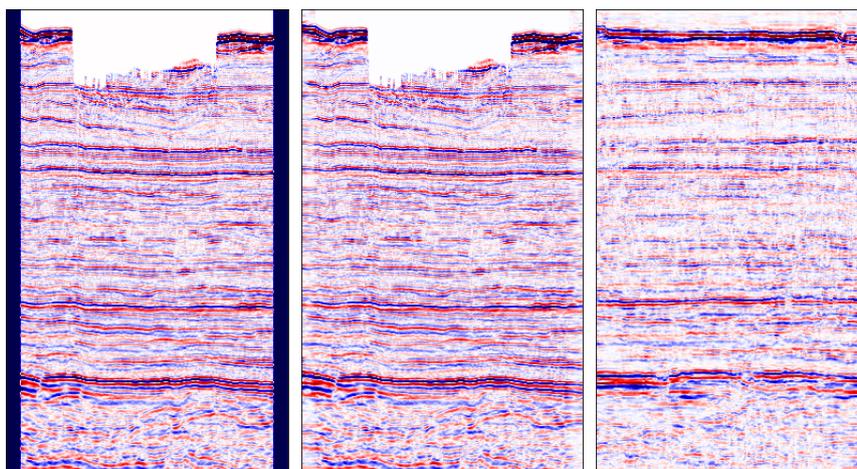


Рис. 11: Пример экстраполяции куба. Слева - инлайн исходного куба, посередине - инлайн расширенного куба, кросслайн, полностью относящийся к предсказанной части

10 эпох с помощью оптимизатора Adam со скоростью обучения $1e-3$.

5.3.2. Результаты

При восстановлении повреждённой части в размере 33 процентов от среза MAE составила 0.16, коэффициент детерминации R2 - 0.91.

Для восстановления каждой второй трассы получены MAE = 0.125, и коэффициент детерминации R2 = 0.92. На основе такой модели было реализовано повышение разрешения куба, позволяющее восстановить исходный куб после удаления части трасс с некоторым шагом, что можно применять для сокращения объёма данных, занимаемого сейсмическим кубом. Однако, такой подход оказался неэффективным по сравнению с суще-

ствующими решениями для сжатия сейсмической информации.

5.4. Сегментация шума

5.4.1. Модель

Использовалась модель U-Net как общепринятая в задачах сегментации. Были опробованы блоки с различными усовершенствованиями. Пробовались блоки с residual ссылками, зарекомендовавшими себя в широком спектре задач. Из-за ограниченности доступного объёма данных рассматривались снижающие число параметров модели слои depthwise separable convolution. Применялись squeeze excitation блоки, эффективность которых в задаче сегментации была показана в [28], в том числе для архитектуры U-Net. Поскольку данные для задачи сегментации были сильно несбалансированы, в качестве функции потерь была использована Dice loss.

5.4.2. Обучение

Из-за ограниченности количества доступных реальных данных модель сначала предобучалась на синтетических данных, а потом обучалась на реальных. Для уменьшения переобучения применялись различные методы регуляризации: weight decay, встроенная в оптимизатор, а также L1 и L2 меры весов, вычисляемые в явном виде и добавляемые к функции потерь. По результатам нескольких запусков была выбрана регуляризация L2. Модели обучались в течение 20 эпох при скорости обучения $1e-2$.

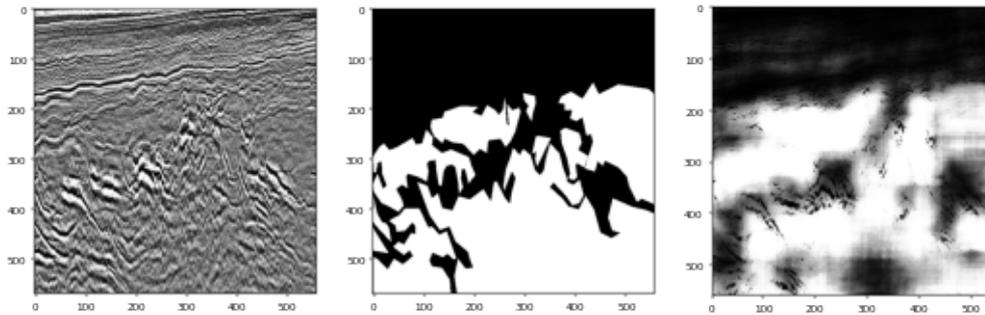


Рис. 12: Пример сегментации. Слева - сейсмический разрез, в центре - маска размеченного шума, справа - сегментация, полученная моделью.

5.4.3. Результаты

Наиболее высокие результаты показали модель с residual ссылками и модель со слоями depthwise separable convolution. На рис. 12 представлен пример сегментации полного изображения сейсмического среза. Метрики представлены в таблице 4. Значительное различие между результатами на валидационной и на тестовой выборках можно объяснить попаданием большого сложного для сегментации изображения в тестовую выборку. Сегментацию можно применять для автоматического поиска проблемных участков, требующих восстановления, но со значительными ограничениями из-за упомянутой выше неоднозначности определения шумных участков.

Таблица 4: Метрики моделей сегментации изображений

Модель	Val F1	Val AUC	Test F1	Test AUC
Базовый U-Net	0.74	0.91	0.34	0.8
Residual	0.76	0.92	0.31	0.77
DWS	0.7	0.91	0.35	0.83
cSE	0.68	0.92	0.32	0.79

Заключение

В рамках данной работы исследовалась возможность применения методов глубокого обучения из области компьютерного зрения для восстановления повреждённых данных сейсмической разведки, а также предсказания отражающих характеристик за пределами исследованной области. Получены результаты:

Реализована сегментация шума на сейсмических срезах. Для обучения модели размечен набор изображений сейсмических срезов. Использована модель U-Net, рассматривались различные блоки: с residual-ссылками, depthwise separable convolution, squeeze excitation модули. Наиболее высокие результаты показали модели с residual ссылками и со слоями Depthwise Separable convolution. Достигнуто значение метрики AUC 0.92 на валидационном датасете и 0.83 на тестовом. Ограничением для применения автоматического выделения шума является неоднозначность его определения.

Реализованы модели для дополнения отсутствующих участков правильной и неправильной формы. Для участков неправильной формы применялась U-Net со слоями gated convolution, которая может обучаться в качестве генератора модели Wasser-

stein GAN. Коэффициент корреляции Пирсона составил 0.69 для простой U-Net и 0.66 для генеративно-сопоставительной модели. Для участков правильной формы использовался простой U-Net с различными слоями: применялись residual ссылки, а также слои gated convolution и partial convolution. Наилучшие результаты показала модель с residual ссылками и слоями partial convolution. Коэффициент корреляции Пирсона составил 0.22 на тестовой выборке и 0.39 на валидационной.

Реализована экстраполяция за пределы исследованной области с помощью модели U-Net со слоями gated convolution, которая может обучаться как генератор Wasserstein GAN. Наилучший коэффициент корреляции Пирсона на тестовом датасете составил 0.53.

Обнаружено, что применение сопоставительного обучения, доминирующего в области дополнения естественных изображений, как и в случае с фотографиями, делает результат визуально более реалистичным, но не повышает метрики, отражающие точность предсказания сейсмических свойств породы по сравнению с классическим обучением с учителем. Дополнительные функции потерь perception loss и style loss также не показали повышения точности предсказания сейсмических разрезов. Для разворачивания решения в промышленных средах требуется дальнейшее усовершенствование.

Голоссарий

Сейсмический куб - трёхмерный массив амплитудных характеристик, получаемых в полевых данных, а также прошедших стадию обработки полевых материалов

Инлайны и кросслайны - срезы сейсмического куба, представляющие собой 2D вертикальные разрезы, где каждый инлайн располагается перпендикулярно каждому кросслайну

Слайс - срез сейсмического куба на определённой глубине (или времени), представляющий собой двумерную карту

Отражающий горизонт - выделенный объект в процессе сейсмической интерпретации, представляющий собой кривую или подошву рассматриваемого пласта

Трасса (сейсмотрасса) - данные, полученные с одного приёмника в поле, представляющие собой одномерный массив амплитудных характеристик. Так, каждый инлайн или кросслайн представляет собой набор сейсмотрасс.

Список литературы

- [1] Arjovsky Martin, Chintala Soumith, Bottou Léon. Wasserstein GAN. — 2017. — 1701.07875.
- [2] Context Encoders: Feature Learning by Inpainting / Deepak Pathak, Philipp Krähenbühl, J. Donahue et al. // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2016. — P. 2536–2544.
- [3] He Kaiming, Zhang Xiangyu, Ren Shaoqing, Sun Jian. Deep Residual Learning for Image Recognition. — 2016.
- [4] He Kaiming, Zhang Xiangyu, Ren Shaoqing, Sun Jian. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. — 2015. — 1502.01852.
- [5] Demir Uğur, Unal Gozde. Patch-Based Image Inpainting with Generative Adversarial Networks. — 2018. — 03.
- [6] Dutta Abhishek, Zisserman Andrew. The VIA Annotation Software for Images, Audio and Video // Proceedings of the 27th ACM International Conference on Multimedia. — MM '19. — ACM, 2019.
- [7] EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning / Kamyar Nazeri, Eric Ng, Tony Joseph et al. // ArXiv. — 2019. — Vol. abs/1901.00212.
- [8] Yu Jiahui, Lin Zhe, Yang Jimei et al. Free-Form Image Inpainting with Gated Convolution. — 2018. — 06.

- [9] Gadylshin Kirill, Silvestrov Ilya, Bakulin Andrey. Inpainting of local wavefront attributes using artificial intelligence for enhancement of massive 3-D pre-stack seismic data // *Geophysical Journal International*. — 2020. — 10. — Vol. 223. — P. 1888–1898.
- [10] Gatys Leon A., Ecker Alexander S., Bethge Matthias. A Neural Algorithm of Artistic Style. — 2015. — 1508.06576.
- [11] Goodfellow Ian J., Pouget-Abadie Jean, Mirza Mehdi et al. Generative Adversarial Networks. — 2014. — 1406.2661.
- [12] High-Resolution Image Inpainting Using Multi-scale Neural Patch Synthesis / C. Yang, X. Lu, Zhe L. Lin et al. // 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2017. — P. 4076–4084.
- [13] Hoorick Basile Van. Image Outpainting and Harmonization using Generative Adversarial Networks // *ArXiv*. — 2019. — Vol. abs/1912.10960.
- [14] Hu Jie, Shen Li, Sun Gang. Squeeze-and-Excitation Networks // 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2018. — P. 7132–7141.
- [15] Iizuka Satoshi, Simo-Serra Edgar, Ishikawa Hiroshi. Globally and locally consistent image completion // *ACM Transactions on Graphics*. — 2017. — 07. — Vol. 36. — P. 1–14.
- [16] Image Inpainting for Irregular Holes Using Partial Convolution

- tions / Guilin Liu, Fitsum Reda, Kevin Shih et al. — 2018. — 04.
- [17] Wang Yi, Tao Xin, Qi Xiaojuan et al. Image Inpainting via Generative Multi-column Convolutional Neural Networks. — 2018. — 10.
- [18] Gulrajani Ishaan, Ahmed Faruk, Arjovsky Martin et al. Improved Training of Wasserstein GANs. — 2017.
- [19] Milletari Fausto, Navab Nassir, Ahmadi Seyed-Ahmad. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. — 2016. — 1606.04797.
- [20] Howard Andrew G., Zhu Menglong, Chen Bo et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. — 2017. — 1704.04861.
- [21] Silva Reinaldo Mozart, Baroni Lais, Ferreira Rodrigo S. et al. Netherlands Dataset: A New Public Dataset for Machine Learning in Seismic Interpretation. — 2019. — 1904.00770.
- [22] Kim Kyunghun, Yun Yeohun, Kang Keon-Woo et al. Painting Outside as Inside: Edge Guided Image Outpainting via Bidirectional Rearrangement with Progressive Step Learning. — 2020. — 2010.01810.
- [23] Baroni Lais, Silva Reinaldo Mozart, Ferreira Rodrigo S. et al. Penobscot Dataset: Fostering Machine Learning Development for Seismic Interpretation. — 2019. — 1903.12060.

- [24] Random Noise Attenuation Using Deep Convolutional Autoencoder / M. Zhang, Y. Liu, M. Bai et al. — 2019. — Vol. 2019, no. 1. — P. 1–5.
- [25] Random noise suppression in seismic data: What can deep learning do? — Vol. All Days of SEG International Exposition and Annual Meeting, 2018.
- [26] Richardson Alan, Feller Caelen. Seismic data denoising and deblending using deep learning. — 2019. — 07.
- [27] Ronneberger Olaf, Fischer Philipp, Brox Thomas. U-Net: Convolutional Networks for Biomedical Image Segmentation. — 2015.
- [28] Roy Abhijit Guha, Navab Nassir, Wachinger Christian. Concurrent Spatial and Channel Squeeze & Excitation in Fully Convolutional Networks. — 2018. — 1803.02579.
- [29] Sabini Mark, Rusak Gili. Painting Outside the Box: Image Outpainting with GANs. — 2018. — 1808.08483.
- [30] Seismic compressive sensing by generative inpainting network: Toward an optimized acquisition survey / Xiaoyang Li, Nikolaos Mitsakos, Ping Lu et al. // The Leading Edge. — 2019. — 12. — Vol. 38. — P. 923–933.
- [31] Seismic data denoising by deep-residual networks. — Vol. All Days of SEG International Exposition and Annual Meeting, 2018.

- [32] Shift-Net: Image Inpainting via Deep Feature Rearrangement / Zhaoyi Yan, Xiaoming Li, Mu Li et al. — 2018. — 01.
- [33] Miyato Takeru, Kataoka Toshiki, Koyama Masanori, Yoshida Yuichi. Spectral Normalization for Generative Adversarial Networks. — 2018. — 1802.05957.
- [34] Using Convolutional Neural Networks for Denoising and Deblending of Marine Seismic Data / Sigmund Slang, Jing Sun, Thomas Elboth et al. — 2019. — 06.
- [35] White noise attenuation of seismic trace by integrating variational mode decomposition with convolutional neural network / Hao Wu, Bo Zhang, Tengfei Lin et al. // Geophysics. — 2019. — Vol. 84, no. 5. — P. V307–V317.
- [36] Xiao Qingguo, Li Guangyao, Chen Qiaochuan. Image Outpainting: Hallucinating Beyond the Image // IEEE Access. — 2020. — Vol. 8. — P. 173576–173583.
- [37] Yu Siwei, Ma Jianwei, Wang Wenlong. Deep learning tutorial for denoising. — 2018. — 10.