Федеральный исследовательский центр «Информатика и управление»
Российской академии наук

На правах рукописи

Достовалова Анастасия Михайловна

Вероятностно-информированные нейросетевые модели анализа изображений при ограниченных обучающих данных

Специальность 1.2.1 —
«Искусственный интеллект и машинное обучение»

Диссертация на соискание учёной степени
кандидата физико-математических наук

Научный руководитель:
доктор физико-математических наук, доцент
Горшенин Андрей Константинович

Москва — 2026

Оглавление

Стр.

Стр.

Стр.

Введение

Актуальность. Извлечение новых знаний из массивов накопленных данных в таких областях, как обработка снимков, получаемых космическими системами и беспилотными летательными аппаратами (БПЛА) [1], молекулярная биология [2] и медицина [3], является одной из важнейших современных исследовательских задач. Часто анализ данных подразумевает распознавание образов или сигналов, включающее в себя математическое моделирование их характерных закономерностей, например, на основе мозаичных представлений однородных участков цифровых изображений [4], или описание специфичной изменчивости с помощью аппарата теории вероятностей и математической статистики [5—7]. Фундаментальные основы таких подходов были заложены в классических трудах К. Пирсона [8], К. Фукунагиа [9], академиков А.Н. Колмогорова [10], А.А. Харкевича [11], Ю.И. Журавлева [12] и К.В. Рудакова [13]. Ряд из них также лег в основу алгоритмов нейронных сетей.

Искусственные нейронные сети (перцептрон) впервые были представлены еще в 1960 годах [14]. Но только в последние десятилетия развитие средств вычислительной техники, алгоритмов оптимизации весов [15], создание более сложных архитектур, таких как сверточные и рекуррентные сети [16—18], позволило эффективно анализировать с их помощью объекты окружающего мира. В условиях, требующих высокой эффективности, адаптивности и точности анализа, алгоритмы искусственного интеллекта (ИИ) оказываются незаменимы.

На сегодняшний день методы машинного обучения и нейронные сети (НС) продемонстрировали значительные успехи в области распознавания образов, в том числе на изображениях [19]. Сверточные сети являются традиционными нейросетевыми архитектурами, применяемыми для классификации и сегментации [20; 21], детектирования объектов [22; 23], в том числе при решении прикладных задач анализа специфических изображений, таких как выделение береговой линии или крон деревьев на спутниковых снимках [24; 25], очагов поражения или физиологических деталей на медицинских изображениях [26] и др.

Обработка данных в сверточных сетях строится на основе операции двумерной свертки с обучаемым ядром $w \in \mathbb{R}^{m \times m}$:

$$y_{i,j} = \sum_{k=-m}^{m} \sum_{l=-m}^{m} w_{m+k,m+l} \cdot x_{i+k,j+l}, \tag{1}$$

где m – размер ядра свертки, $X = \{x_{i,j}\}, i = \overline{1,W}, j = \overline{1,H}$ – исходный снимок, а $Y = \{y_{i,j}\}$ – изображение, полученное в результате свертки. С момента появления сверточные архитектуры претерпели множество модификаций. Были разработаны более сложные формы свертки. Так, в архитектурах семейства DeepLab [27] для увеличения рецептивного поля была реализована агрегация признаков с помощью расширенных свёрток и пространственного пирамидального пулинга. Модификации позволили повысить точность распознавания сложных сцен и увеличить масштаб обработки изображения – с 32×32 до 2048×2048 пикселей.

Основным недостатком сверточных НС является их ограниченная способность понимания глобальных взаимосвязей между элементами изображения, что сильно осложняет анализ с их помощью снимков высокого разрешения. Альтернативой сверточным сетям стали трансформерные архитектуры. В сети Visual Transformer (ViT) [28] изображение впервые было представлено как набор токенов, связи между которыми обрабатывались с использованием механизма самовнимания (англ. self-attention) для выявления наиболее важных связей между элементами векторов данных. Применение к вектору $\mathbf{x}$ внимания описывается формулой:

$$Attention(\mathbf{x}) = softmax(\frac{\mathbf{x} \cdot W_q \cdot W_k^T \cdot \mathbf{x}^T}{\sqrt{d_k}})\mathbf{x} \cdot W_v, \tag{2}$$

где W_q – матрица линейного преобразования вектора $\mathbf{x}$ для получения вектора-запроса, W_k – матрица для вектора-ключей, W_v – матрица вектора-значений, d_k – размерность вектора ключей. Механизм внимания позволяет выделить наиболее важные связи между элементами вектора $\mathbf{x}$.

Трансформеры были адаптированы для решения задач сегментации (архитектура Segmenter [29] – 2021 г.), детекции [30] и др., а также и для обработки специализированных типов изображений [1; 31; 32]. Однако главными недостатками трансформерных архитектур являются вычислительная сложность и потребность в большом количестве обучающих данных. Множество их модификаций были направленны на повышение производительности сети: например, в Data-efficient Image Transformer применяется механизм дистилляции внимания [33], в архитектуре Refiner используются проекции матриц внимания [34], а в работах [35] и [36] используются специализированные блоки кросс-патчевого и легковесного группового внимания, сокращающие вычисления до обработки

наиболее значимых элементов. Часть модификаций для повышения интерпретируемости результата внедряют в трансформеры элементы сверточных или графовых сетей [37—39], например, для вычисления эмбеддингов (Conditional Position-encoding Vision Transformer [40]) или формирования токенов с учетом перекрытия рецептивных полей (Swin transformer [41]).

Однако в прикладных и научных задачах, например, в сфере наук о материалах, биологии или обработке спутниковых изображений [2; 42; 43], а также в медицинских приложениях [44], сложные модифицированные сверточные или трансформерные архитектуры [30; 39] нередко демонстрируют значительно более слабые результаты, чем при анализе обычных данных. Причиной тому являются характерные особенности предметной области, такие как сложность проведения и обработки результатов наблюдений и экспериментов или редкость исследуемых явлений, из-за которых доступные обучающие наборы данных оказываются ограниченными по некоторым характеристикам – а именно являются несбалансированными, малыми по числу элементов или сильно неоднородными.

Несбалансированность обучающего набора означает, что элементов некоторой категории объектов в нем существенно меньше, чем остальных. Примером таких данных являются изображения, содержащие малоразмерные [45] или фоновые объекты [46] – на панорамах улиц или аэрокосмических (спутниковых и БПЛА) снимках поверхности Земли. При этом несбалансированный набор может содержать довольно большое количество наблюдений [47]. Малым же набор называется, если содержит всего порядка нескольких сотен или тысяч наблюдений [43] (в литературе обычно границы определены нечетко и изменяются от 100 до 2-3 тысяч элементов). Наконец, неоднородными или изменчивыми называют наборы с выраженной внутриклассовой дисперсией. Неоднородность может быть обусловлена особенностями источников данных или оборудования [48], как, например, географическая и суточная изменчивость при аэрокосмической съемке. Нередко неоднородность является следствием ограниченности и несбалансированности обучающего набора.

Общей чертой неоднородных, несбалансированных и малых по числу элементов датасетов является отсутствие достаточного количества информативных признаков для построения правильных закономерностей. Строящееся на таком наборе решающее правило не воспроизводит вариативность реальных данных, что, аналогично случаю малых выборок в статистике, повышает подверженность правила к искажениям и переобучению [49]. При этом, если ис-

ходный датасет специфичен, универсальная стратегия добавления данных из открытых датасетов является неэффективной. Объединение данных, различных по свойствам, зачастую приводит к усилению внутриклассовой дисперсии [50], особенно если исходный датасет ограничен.

Проблема построения решающих правил в условиях ограниченности обучающих признаков является характерной для научных и технических задач, в которых требуется разработка высокоточных и надежных, доверенных [51], методов ИИ. Высокая потребность в автоматизации обработки ограниченных сложных технических датасетов привела к созданию множества методов, альтернативных традиционному расширению набора открытыми данными. Наиболее популярными подходами для повышения точности обработки ограниченных данных являются методы семплинга [52], специальные алгоритмы аугментации [53], целенаправленно увеличивающие количество обучающих примеров малочисленных классов, и даже генеративные модели [54; 55]. Развиты различные техники регуляризации [56], в том числе фокальной (англ. focal loss) и со взвешиванием классов [57—59]. Перспективными являются архитектурные модификации, такие как специальные модули внимания [60], блоки обработки иерархических [61; 62], глобальных и локальных признаков данных [63] или графово-сверточные решения [64; 65]. Наконец, применяются методы переноса обучения [66; 67], настраиваемые на больших и разнообразных открытых датасетах. Природа данных для предобучения иногда может отличаться от исходных: например, для обработки радиолокационных снимков могут использоваться оптические изображения тех же объектов съемки [68]. Возможно также предобучение на данных другой модальности [69], например, между данными биологических последовательностей РНК и ДНК [70].

В научных и технических задачах в качестве отдельного направления выделяют подход информирования, предлагающий использовать математические модели данных при обучении нейронных сетей для учета особенностей и ограничений предметной области. Впервые идея привлечения математических моделей для повышения эффективности методов искусственного интеллекта возникла в 90-х годах прошлого века в работах [71; 72], где нейросети использовались для решения обыкновенных дифференциальных уравнений и уравнений в частных производных, решение которых представлялось в виде нейросети, структура которой выбиралась так, чтобы граничные условия удовлетворялись автоматически. Термины же информирования и физически-

информированных нейронных сетей ввел Дж. Карниадакис в 2017-2019 годах, формализовав подход и сделав его тем самым более удобным для практического использования [73]. Математические модели или выполняют роль источника дополнительных признаков, или вводят ограничения на область поиска решения, справедливые для предметной области, которые в условиях ограниченных наборов не могут быть выведены напрямую из данных. Следует отличать информирование от объединения НС с моделями, при котором НС выступает в качестве численного метода оценки параметров. Например, в работе [74] вариационный автокодировщик использовался для аппроксимации распределения скрытых переменных модели факторного анализатора [75]. При информировании модель является источником дополнительных данных, а кроме того, становится возможным решение обратных задач без принципиального изменения архитектуры сети.

По способу реализации выделяют три основных способа информирования нейросетей (рисунок 1). При информировании на уровне признаков вектор входных данных расширяется его характеристиками, полученными из математической модели. При информировании на уровне функции потерь к стандартным обучающим функционалам добавляются регуляризующие слагаемые, гарантирующие выполнение свойств модели, например, граничных условий. Наконец, при информировании на уровне архитектуры в сеть добавляется специальный блок, в котором этапы обработки данных математической моделью повторяются с помощью нейросетевых слоев. При этом могут также использоваться уникальные параметры модели, например, матрицы переходных вероятностей.

Подход информирования продемонстрировал высокую эффективность в задачах математической физики, требующих решения систем дифференциальных уравнений в частных производных – Бюргерса, Курамото-Сивашинского, Навье-Стокса и др. [76—78]. Успехи продемонстрированы в области гидродинамики [79; 80] и динамики неэластичных жидкостей [81], а также для моделирования климата, землетрясений, тепловых и сверхзвуковых потоков [82—84], колебательных движений [85], и оценке надежности конструкций [86] и медицине – например, в работе [87] снимки магнитно-резонансной томографии использовались для оценки церебральной атрофии, описываемой обыкновенным дифференциальным оператором первого порядка. Важным направлением является применение математических моделей для повышения интерпрети-

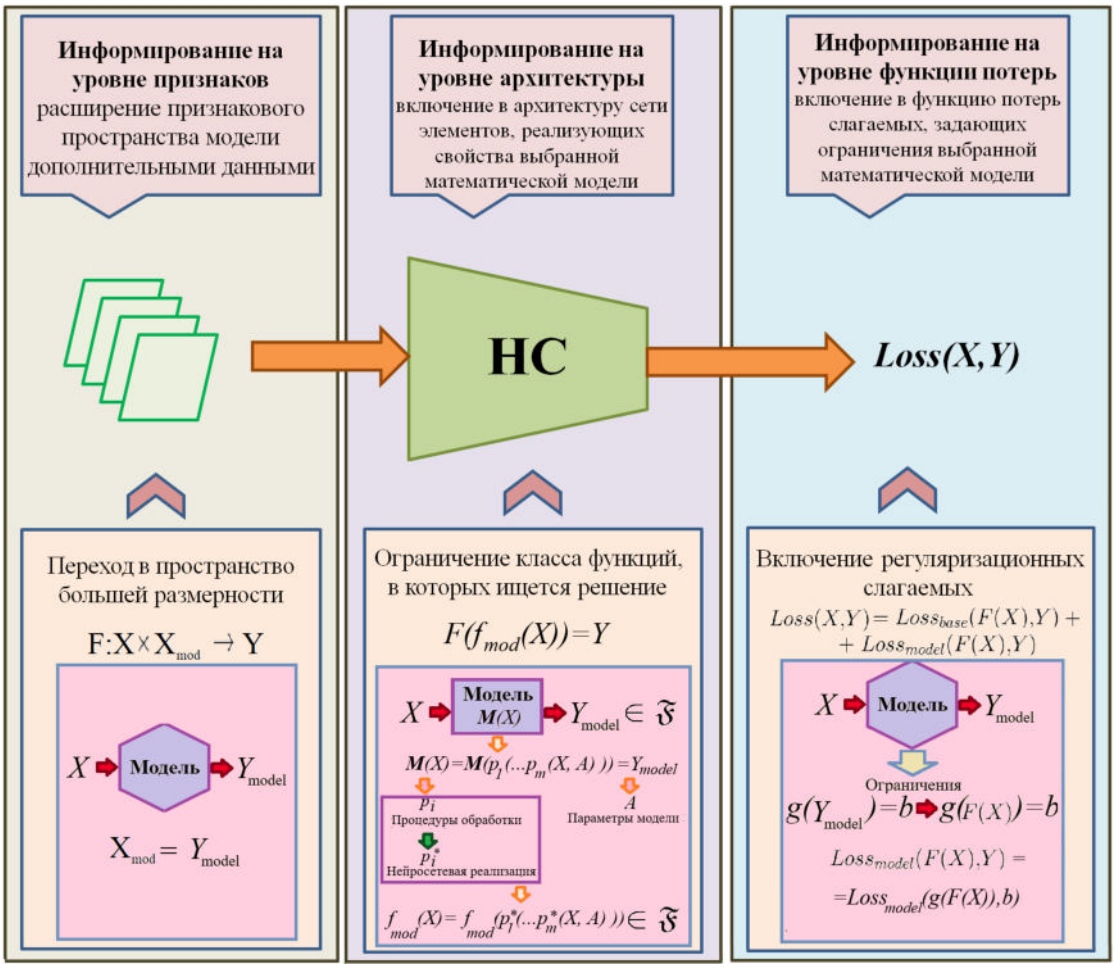

Рисунок 1 — Способы реализации информирования нейронных сетей

руемости результатов нейронных сетей, в том числе аналитической оценке точности получаемых прогнозов [88] или разработке теоретической основы для восстановления вознаграждения, которое объясняет поведение эксперта [89].

В задачах обработки изображений применение подхода информирования сопряжено со сложностями, из-за отсутствия для них в общем случае универсальных адекватных физических моделей. Известные применения подхода, как правило, реализованы на уровне признаков: используются спектральные представления снимков, полученные преобразованием Фурье [90—92], условия фотометрической согласованности [90] и геометрические характеристики объектов на изображениях [93], такие как их расположение [94], азимутальный угол [95] или даже линейные сплайновые представления контуров [96].

Однако для изображений как многомерных сигналов зачастую справедливыми оказываются вероятностно-статистические представления, например, описывающие поведение шумовой составляющей сигнала. Информирование сетей такими вероятностными моделями продемонстрировало успехи в задачах обработки последовательностей и временных рядов, например при оценке

неопределенности предсказаний [97; 98], надежности инженерных систем и функций риска [99], уточнения оценок функции стоимости [100] или улучшения интерпретируемости прогнозов нейронных сетей с пиковыми нагрузками [101] (англ. spiking neural networks). Аналогичный подход применяется для повышения эффективности предсказаний поведения динамических систем [102; 103]. Обычно используемые модели помех зачастую оказываются общими, как, например, в диффузионных или байесовских архитектурах [104; 105]. Вероятностные модели также могут описывать пространство скрытых признаков [106] или процесс их слияния [107] – обучение параметров моделей может быть отделено от процесса обновления весов нейронной сети [108; 109]. Информирование моделью гауссовского процесса повышает точность предсказания вида кривых «вероятность-напряжение-долговечность» для оценки эксплуатационных характеристик зубчатых передач [110], а также деталей, изготовленных методом быстрого прототипирования [111].

Как демонстрируют работы А.К. Горшенина [112—116], моделирование структурных особенностей стохастических процессов с помощью модели конечной смеси вероятностных распределений (формирования на ее основе компонент связности временного ряда) и использование ее характеристик (значений математического ожидания, дисперсии, коэффициента эксцесса и др.) в качестве дополнительных входных признаков существенно повышает точность предсказаний нейронных сетей, в том числе для анализа малых наборов – данных о состоянии турбулентной плазмы и информационных потоков. Встраивание элементов вероятностной модели (например, глубокой гауссовской смеси [117]) в архитектуру нейронной сети также демонстрирует эффективность в задачах предсказания трафика и иных временных рядов [118; 119].

Основным преимуществом похода информирования является его большая универсальность и применимость для сильно ограниченных и специфичных технических данных в сравнении с альтернативными методами. Например, методы переноса обучения часто сталкиваются с трудностями из-за различия предметных областей базового и целевого наборов данных, кроме того, часто они требуют использования дополнительных метаданных, которые для целевого набора не всегда существуют. Дополнение набора синтетическими данными (аугментация, семплинг и др.) решает задачу перераспределения вероятностей признаков, выделенных нейронной сетью, и сталкивается с проблемой воспроизведения вариативности реальных данных [120; 121]. По этой причине задача

развития методов информирования нейронных сетей для обработки ограниченных наборов изображений является актуальной.

Целью данного научного исследования является развитие развитие теоретически обоснованных вероятностно-информированных нейросетевых моделей для решения задач обработки неоднородных, несбалансированных и малых наборов изображений.

Для достижения поставленной цели необходимо было решить следующие **задачи**:

1. Развить теоретические основы, определяющие выбор вероятностной модели и способа информирования в задачах обработки ограниченных наборов изображений.
2. Выполнить аналитическое исследование разработанных моделей и информированных архитектурных блоков нейронных сетей в условиях ограниченных данных.
3. Разработать методы информирования вероятностными моделями различных нейросетевых архитектур.
4. Апробировать разработанные подходы для решения задач классификации и сегментации несбалансированных, малых и неоднородных наборов изображений.

Методология и методы исследования. В диссертации предложены оригинальные подходы и процедуры, в том числе:

- метод информирования нейронных сетей с помощью дополнительных входных признаков, построенных с помощью смешанных вероятностных моделей для повышения точности сегментации неоднородных наборов изображений;
- метод информирования нейронных сетей композицией вероятностных моделей конечной смеси распределений и случайного поля Маркова для повышения точности сегментации малых и неоднородных наборов изображений;
- метод архитектурного информирования моделью факторного анализатора построения блока слияния признаков для повышения точности классификации ограниченных по числу элементов наборов изображений;

– метод архитектурного информирования моделью случайного Марковского поля для сегментации сильно несбалансированных наборов изображений
– метод повышения точности выделения на изображениях малоразмерных объектов с помощью информированных графовых ансаблей .

Применяются и такие классические методы исследования, как аналитический аппарат теории вероятностей и математической статистики для смешанных распределений, методы параметрического и непараметрического статистического оценивания, проверка статистических гипотез, методы линейной алгебры, в том числе для оценки сложности предлагаемых вычислительных процедур, алгоритмы машинного обучения и нейронные сети.

Для создания комплекса программных решений, предназначенных для анализа данных, использованы языки программирования C++ и Python, а также современные высокопроизводительные вычислительные ресурсы ЦКП «Информатика» ФИЦ ИУ РАН.

Научная новизна: В диссертации впервые выполнена адаптация подходов вероятностного информирования нейронных сетей для решения задач обработки изображений, включающая:

1. теоретическое доказательство свойств вероятностных моделей изображений, включающих случайное поле Маркова в виде квадродерева и факторный анализатор с импульсно-аддитивным шумом, обосновывающих выбор способа информирования и возможность повышения с их помощью точности обработки ограниченных наборов данных;
2. теоретическое доказательство свойств информированных архитектурных блоков, включающих оценку вычислительной сложности, повышение точности обработки малых наборов и ускорение обучения сети;
3. теоретически обоснованный метод архитектурного информирования моделью случайного Марковского поля графовых нейросетей для повышения точности обработки несбалансированных наборов изображений;
4. теоретически обоснованный метод архитектурного информирования моделью факторного анализатора сверточных нейросетей для повышения точности классификации малых наборов изображений;
5. теоретически обоснованный метод информирования композицией вероятностных моделей смеси и Марковского случайного поля для учета

признаков разной природы для обработки сильно неоднородных наборов данных;

Основные положения, выносимые на защиту:

1. Метод архитектурного информирования моделью Марковского случайного поля нейронных сетей с доказательством теорем о более быстром убывании функции потерь;
2. Метод архитектурного информирования моделью факторного анализатора с импульсно-аддитивным шумом в блоке слияния признаков с доказательством теорем о его аналитических свойствах и оценке вычислительной сложности;
3. Метод комбинированного информирования на уровне признаков и архитектуры сети композицией моделей конечной смеси вероятностных распределений и случайного поля Маркова с доказательством теоремы о повышении точности обработки неоднородных наборов данных;
4. Аналитические свойства модели Марковского случайного поля в виде квадродерева, в том числе теорема о связи с графовыми нейронными сетями.

Практическая значимость Результаты диссертации являются одновременно фундаментальными и прикладными. Эффективность разработанных методов анализа данных и вычислительных процедур определяется полученными в диссертации математическими результатами, включающими обоснование выбора способа информирования сети, оценки вычислительной сложности и скорости обучения вероятностно-информированных нейросетевых блоков. Эти результаты подтверждаются и путем всестороннего тестирования созданных методов на реальных ограниченных наборах изображений, что продемонстрировано в диссертации на примерах анализа аэрокосмических снимков земной поверхности (полученных как с помощью спутников, так и БПЛА) и открытых наборов обычных изображений.

Достоверность полученных результатов обеспечивается аналитическими доказательствами свойств предлагаемых методов и обоснованием их эффективности для работы с ограниченными наборами данных; всесторонним эмпирическим тестированием предложенных методов на разнообразных по свойствам открытых наборах данных, а также соответствием с результатами, полученными другими авторами.

Апробация работы. Основные результаты работы докладывались на следующих научных мероприятиях:

- Международная конференция по искусственному интеллекту AI Journey: 2024 [124], 2025 гг. [126];
- Международная конференция «Интеллектуальные системы» (INTELS'24): 2024 г. [130];
- Конференция молодых учёных «Фундаментальные и прикладные космические исследования»: 2025 г;
- Международная научная конференция студентов, аспирантов и молодых учёных «Ломоносов»: 2024 [134], 2025 гг. [133];
- Научная конференция «Ломоносовские чтения»: 2025 г. [131];
- Научная конференция «Тихоновские чтения»: 2024 г. [132];
- Научный семинар факультета компьютерных наук НИУ ВШЭ под руководством профессора А.А. Наумова: 2026 г.;
- Научный семинар кафедры математической статистики ВМК МГУ «Интеллектуальные методы вычислительной статистики»: 2023-2025 гг.
- Научный семинар кафедры математической статистики ВМК МГУ «Теория риска и смежные вопросы»: 2026 г.;
- Научные семинарах отделений 1, 5 и 6 ФИЦ ИУ РАН: 2025-2026 гг.

Основные результаты диссертации были получены в процессе выполнения работ по гранту №075-15-2024-544 «Математические модели и численные методы как основа для разработки робототехнических комплексов, новых материалов и интеллектуальных технологий конструирования» Министерства науки и высшего образования РФ.

Публикации. Основные результаты по теме диссертации изложены в 13 печатных изданиях, 7 из которых изданы в журналах, рекомендованных ВАК, 9 — в периодических научных журналах, индексируемых Web of Science и Scopus, 4 — в тезисах докладов.

Личный вклад. Основные результаты диссертации получены автором самостоятельно. В работах [123—125] А.М. Достоваловой разработаны подходы вероятностного информирования для решения задач обработки неоднородных и несбалансированных наборов изображений, а также проведен всесторонний анализ полученных результатов. В работах [122—128; 130] А.М. Достоваловой доказан ряд теорем, обосновывающих эффективность предлагаемых способов информирования сетей в конкретных задачах, а также развиты и исследованы

математические модели, методы и вычислительные алгоритмы анализа неоднородных и несбалансированных данных с реализацией в виде программных решений и их приложениями к обработке аэрокосмических изображений Земной поверхности.

Кратко остановимся на содержании работы. Диссертация состоит из трех глав, каждая из которых посвящена решению задачи обработки одного из типов ограниченных наборов изображений – малого, неоднородного или несбалансированного датасета.

Первая глава посвящена информированной нейросетевой модели слияния глобальных признаков изображений для повышения точности классификации малых наборов. Для моделирования процесса слияния в разделе 1.2 была представлена новая вероятностная модель факторного анализатора с импульсным и аддитивным шумами (формула (1.3)). Предполагается, что импульсная помеха полностью искажает полезную информацию в сигнале и все искаженные элементы могут быть приравнены к постоянному значению y_0:

$$Z = \eta \cdot (A \cdot \Theta + \xi_0) + y_0 \cdot (I - \eta),$$

где Z – наблюдаемое значение, η – матрица, на диагоналях которой расположены независимые одинаково распределенные биномиальные случайные величины с вероятностью успеха d_{prob}, описывающие импульсную помеху, I – единичная матрица, A – матрица нагрузок, Θ – матрица факторов модели и ξ_0 – стандартная нормально распределенная случайная величина, описывающая аддитивную помеху.

Модель (1.3) позволяет одновременно учитывать часто встречающиеся на изображениях искажения, такие как импульсные и аддитивные помехи. Для модели были исследованы ее математические свойства. Доказана теорема 1, утверждающая, что параметры факторного анализатора с аддитивным и импульсным шумами могут быть определены однозначно. В теореме 2 определены ограничения на количество факторов модели, следующие из теоремы Андерсона-Рубина [75]. Теорема 3, представленная в этом же разделе, устанавливает, что оценки параметров модели факторного анализатора с аддитивным и импульсным шумами, полученные в ходе минимизации кросс-энтропии, являются несмещенными и состоятельными, что значит, что они не накапливают систематические искажения. Теоремы 1-3 обосновывают выбор модели факторного

анализатора с аддитивным и импульсным шумами для информирования при обработке малых наборов.

В разделе 1.3 была представлена архитектура сети Factor Fusion Neural Network (FtFNN), информированной моделью факторного анализатора с аддитивным и импульсным шумами. Поскольку модель использовалась для объединения глобальных признаков снимка, информирование было реализовано на уровне архитектуры сети. FtFNN состоит из кодировщика и информированного классификатора $G(\cdot)$, реализующего слияние глобальных признаков снимка в разных пространственных разрешениях, которые рассматриваются в качестве факторов модели. Обработка представлений снимка в $G(\cdot)$ разделена на три этапа, повторяющих этапы обработки факторным анализатором (подробное описание представлено в разделах 1.3.1-1.3.3):

– отображение в пространство факторов;
– учет шумовой составляющей;
– умножение на матрицу нагрузок.

В разделе 1.4.2 доказано (теорема 4), что при определенных значениях гиперпараметров классификатор FtFNN оказывается вычислительно проще аналогов, построенных в виде композиции полносвязных и сверточных слоев

В разделе 1.4 представлены результаты обработки FtFNN 13 малых датасетов. FtFNN сравнивалась с семью сверточными архитектурами для классификации изображений, такими как EfficientNet B0 [135], Xception [136] и др. (трансформерные сети не рассматривались из-за сильной ограниченности обучающих данных). Кодировщики признаков этих архитектур использовались в качестве вариантов реализации кодировщика FtFNN.

FtFNN превосходит по точности все рассмотренные базовые сети. Максимальные приросты классификационных метрик Top-1, Top-3 и Top-5 Accuracy составляют 16.9%, 10.23% и 5.67%. Средний прирост Top-1 Accuracy, полученный при перекрестной проверке, достигает 14.4%. Показано, что согласно статистическому критерию Фридмана разница в значениях Top-1 Accuracy является статистически значимой при уровне значимости 0.01. Также установлено, что прирост точности классификации тем выше, чем меньше элементов в обучающем наборе и чем более данные неоднородны.

Моделирование шума в FtFNN повышает точность классификации над базовыми архитектурами: в 45 из 72 тестов при моделировании аддитивного шума, а в 65 из 72 – импульсного. Меньшая вычислительная сложность

классификатора FtFNN продемонстрирована экспериментально (подтверждение теоремы 4) – уменьшение числа выполняемых операций достигает 1.3 MFLOPS. Количество параметров сети увеличивается по сравнению с базовым классификатором только в 10 из 72 тестов, в остальных случаях оно уменьшается – максимальное снижение достигает 496 тысяч.

Вторая глава посвящена нейросетевой модели сегментации сильно неоднородных наборов изображений, информированной композицией вероятностных моделей. Поскольку рассматривалась задача сегментации, были выбраны модели, описывающие локальные характеристики снимка: конечная смесь нормальных законов и случайное поле Маркова в форме квадродерева: K-компонентная смесь нормальных распределений [137] (K – число выделяемых классов) моделирует яркости отдельных пикселей снимка, тогда как поле Маркова – пространственные взаимосвязи между ними.

В модели смеси для каждого пикселя яркостью X_i вычисляется вектор $p^*(X_i) = \left(p_1\,\varphi\left(\frac{X_i-a_1}{\sigma_1}\right),\ldots,p_K\,\varphi\left(\frac{X_i-a_K}{\sigma_K}\right)\right)$ вероятностей соответствия каждой из K компонент смеси нормальных распределений с плотностью $\varphi(\cdot)$ (a_j и σ_j – параметры сдвига и масштаба компонент смеси). В разделе 2.2 доказана теорема 5, демонстрирующая, что информирование смесью должно быть реализовано на уровне входных признаков, поскольку если целевой набор данных неоднороден, такой способ информирования вероятностями компонент смеси позволяет уменьшить ошибку восстановления целевой функции по сравнению с сетью без информирования.

В разделе 2.3 представлено описание случайного поля Маркова в форме квадродерева [138], моделирующего пространственные взаимосвязи между пикселями. Квадродерево можно представить в виде пирамиды изображений разного пространственного разрешения $S_0 \ldots S_{h-1}$, где S_0 - нижний слой дерева. Каждый узел $s \in S_l$ соответствует определенному пикселю в пирамиде изображений и связан с одним родительским узлом s^- в предыдущем (верхнем) слое и с четырьмя дочерними узлами s^+ в следующем слое. Также для каждого узла дополнительно определяется множество $\{s^* \in S_l, s \in S_l, s^{\preceq} s\}$, состоящие из ближайших соседних узлов s в слое S_l. Таким образом, квадродерево представляет собой структуру, одновременно описывающую пространственные и иерархические взаимосвязи между элементами изображения.

В разделе 2.3 доказываются свойства модели. Согласно теореме 6, обработка изображений с использованием квадродерева может быть интерпретирована

как обработка графово-сверточной нейронной сетью. Поэтому информирование этой моделью следует реализовать на уровне архитектуры сети. Кроме того, поле Маркова в виде пространственно-иерархического квадродерева является эргодичным (теорема 7).

В разделе 2.4 представлено описание созданной на основе теорем 5-6 информированной композицией моделей архитектуры Probability Informed Neural Network (PrINN). Сеть состоит из трех блоков: блока моделирования входных изображений моделью смеси, блока сегментации базовой сверточной или трансформерной сетью, и блока пост-обработки с помощью квадродерева. В разделах 2.5.3-2.5.6 представлены результаты тестирования PrINN на семи датасетах, построенных на основе изображений, полученных радиолокаторами Sentinel-1, ESAR, Capella, и снимка из набора HRSID. Обучающие наборы ограничены по размеру и сильно неоднородны, поскольку из-за зашумленности и особенностей рельефа поверхностей данные характеризуются малыми межклассовыми различиями.

PrINN существенно превосходит по точности все рассмотренные базовые сверточные и трансформерные сети без информирования: прирост точности достигает 20.31% по метрике Accuracy и 19.24% – по метрике F_1. Согласно результатам, представленным в разделе 2.5.7, информирование композицией двух вероятностных моделей значительнее повышает точность обработки неоднородных наборов в сравнении с информированием каждой моделью по отдельности. Так для сетей, информированных только конечной смесью, прирост среднего значения Accuracy относительно результатов сетей без информирования достигает 8.02%, а F_1 – до 18.14%. Для сетей, информированных только квадродереввом прирост Accuracy достигает 5.4%, а F_1 – до 5.89%.

Третья глава посвящена разработке информированной НС модели для повышения точности обработки пространственных связей между внутренними признаками снимка в нейросетевом сегментаторе в условиях сильно несбалансированных наборов. В качестве примера такой задачи рассмотрена сегментация снимков, включающих разномасштабные и малоразмерные объекты.

Эргодичность поля Маркова в виде квадродерева (см. теорему 7) позволяет переносить закономерности, выделенные для крупных объектов в слоях квадродерева $S_1, \ldots, S_{h-1}$ на малые объекты в слоях более высокого разрешения и наоборот. При этом, согласно теореме 6, доказанной в главе 2, информирова-

ние квадродеревом следует реализовать на уровне архитектуры сети с помощью обучаемых графово-сверточных слоев.

В разделе 3.2 представлены свойства нейросетевой модели, информированной квадродеревом. Доказана теорема 8 о том, информированный квадродеревом графовый блок может обучаться быстрее, чем архитектура, выполняющая графовую свертку по двумерной решетке, традиционно применяющейся для моделирования изображений.

Для обработки снимков высокого разрешения в разделе 3.2 был создан новый двухветочный графовый блок. Одна его ветвь обрабатывает глобальные признаки входного снимка с помощью информированного квадродеревом блока, а вторая – локальные взаимосвязи внутри подобласти (суперпикселя), моделируемые двумерной решеткой. Доказаны теоремы 9 и 10, демонстрирующие, что двухветочный графовый блок может обучаться быстрее, чем сопоставимые линейные графовые и сверточные сети, в том числе с вниманием.

В разделе 3.3 представлена созданная на основе теорем 6-10 информированная архитектура FN-QiGSAN с двухветочным графовым блоком, которая также включает сверточный или трансформерный кодировщик, блок сжатия входных признаков кодировщика и блок объединения результатов двух ветвей в выходное изображение. В разделе 3.4 представлены результаты тестирования FN-QiGSAN в задачах двухклассовой и многоклассовой сегментации четырех наборов аэрокосмических снимков высокого разрешения – HRSID, SSDD, UDD и UAVid. В качестве кодировщика рассматривались различные сверточные (DeepLabV3 [27], ENet [139], PSPNet [140] и др.) и трансформерные (SegFormer [141], LWGANet [36] и др.) архитектуры.

Результаты многоклассовой сегментации продемонстрировали, что FN-QiGSAN повышает точность обработки изображений в сравнении со всеми базовыми сегментаторами. Средний прирост значений F_1-меры для крупных объектов относительно трансформеров составляет 6.58%, а относительно сверточных сетей – 14.67%. Для малых объектов (автомобили) средний прирост F_1-меры составляет 9.36% относительно трансформеров и 11.89% – относительно сверточных сетей. FN-QiGSAN демонстрирует существенное улучшение точности обработки изображений в сравнении с SOTA-моделью LWGANet – для отдельных классов крупных объектов прирост точности распознавания достигает 15.11%. FN-QiGSAN также демонстрирует более высокую точность сегментации в сравнении с альтернативными реализациями графового ансам-

бля, прирост точности у которых в среднем составляет всего 6.5% относительно базовых сегментаторов.

В задаче двухклассовой сегментации FN-QiGSAN также демонстрирует превосходящую точность выделения малоразмерных объектов в сравнении со всеми рассмотренными базовыми архитектурами. Прирост точности сегментации кораблей по F_1-мере достигает 66.24% в сравнении с трансформерными сетями, и 62.05% в сравнении со сверточными. Прирост точности сегментации автомобилей на БПЛА изображениях достигает 32.81% по F_1-мере в сравнении с трансформерами, и 23.76% – в сравнении со сверточными архитектурами.

Лучшие по точности конфигурации FN-QiGSAN демонстрирует результаты превосходящей или сопоставимой с трансформерами (SegFormer и LWGANet) точности при использовании простых сверточных кодировщиков, например U-Net, ENet или DeeplabV3, а уменьшение числа параметров сети достигает 1.78 раз при многоклассовой сегментации, и до 13.4 раз – при двухклассовой.

Благодарности. Автор выражает искреннюю признательность своему научному руководителю доктору физико-математических наук, доценту Андрею Константиновичу Горшенину за полезные обсуждения, ценные рекомендации и плодотворные совместные исследования.

Объем и структура работы. Диссертация состоит из введения, 3 глав и заключения. Полный объём диссертации составляет 160 страниц, включая 40 рисунков и 41 таблицу. Список литературы содержит 231 наименование.

Глава 1. Нейросетевые классификаторы изображений, информированные факторными анализаторами

Обработка малых наборов изображений является важной прикладной задачей в области медицины и технологических процессов [142—144]. Ограниченность обучающего набора может быть вызвана высокой стоимостью получения наблюдений, трудоемкостью разметки, а также редкостью наблюдаемых явлений – например, количество пациентов с определенным заболеванием может исчисляться десятками или сотнями [145]. Предварительная обработка и разметка специфических данных требует экспертных знаний, что также ограничивает объем обучающего набора. Например, для создания датасета LiTS [146] для диагностики поражений печени по ее изображениям, полученным магнитно-резонансной томографией (МРТ), потребовались усилия специалистов из семи стран, которые имели не менее трех лет опыта работы в данной сфере.

Недостаток обучающих данных вызывает необходимость привлечения дополнительной информации о классифицируемых элементах. Поэтому большое количество методов обработки малых наборов включают в себя техники объединения (слияния) глобальных (т.е. соответствующие всему изображению, а не его отдельным частям) признаков: как, например, данных другой модальности, так и преобразований изображения, например, их представлений в различных пространственных разрешениях, сформированных с помощью пулинга [147] или других нейронных сетей [148].

Основными техниками слияния признаков являются суммирование и конкатенация [149], в том числе с ранжированием признаков с использованием обучаемых весов [150—152]. Однако они не учитывают стохастические свойства данных – в особенности, искажения различными типами помех. В главе рассматривается новый подход к созданию НС модели слияния признаков, информированной факторным анализатором. По определению [75] эта модель выполняет предсказание за счет объединения информации из нескольких наблюдаемых зашумленных факторов. Факторные анализаторы придают свойство локальной линейности признаковому пространству [153]. С одной стороны, это способствует его расширению, как и учет влияния шума, а с другой, делает его структуру более определенной, способствуя эффективному восстановлению взаимосвязей в условиях малых наборов. В главе доказываются свойства моде-

ли факторного анализатора, модифицированного для работы с зашумленными изображениями, обосновывающие эффективность информирования нейроных сетей этой моделью в условиях недостатка обучающих данных. Также аналитически определяется выбор способа информирования этой моделью нейронной сети.

1.1 Постановка задачи

Пусть дан малый набор (количество элементов набора N около 1000) зашумленных изображений $X = \{X_i\}_{i=\overline{1,N}}$, $X_i \in \mathbb{R}^{H_X \times W_X}$, и $ns(\cdot)$ – функция, описывающая искажение. Для повышения информативности набора X каждому изображению X_i ставятся в соответствие M_0 дополнительных глобальных признаков $\{X_i^{(l)}\}_{l=\overline{1,M_0}}$.

Ставится задача разработать информированную НС модель слияния признаков $\{X_i^{(l)}\}_{l=\overline{1,M_0}}$ с учетом $ns(\cdot)$ $G(\cdot)$ вида:

$$G : \mathbb{R}^{X^{(1)}} \times \ldots \times \mathbb{R}^{X^{(M_0)}} \to \mathbb{R}^K, \tag{1.1}$$

для повышения вероятности правильной классификации на K классов набора X сетью $f_X(\cdot) = H(Enc(\cdot))$ (где $H(\cdot)$ – классификатор):

$$\mathbb{P}(G(ns(Enc(X_i))) = Y^i) \geqslant \mathbb{P}(f_X(X_i) = Y^i).$$

Схема НС модели, рассматриваемой в решаемой задаче, представлена на рисунке 1.1.

В качестве $\{X_i^{(l)}\}_{l=\overline{1,M_0}}$ рассматриваются признаки изображения в разных пространственных разрешениях $X_i^{(l)} \in \mathbb{R}^{X^{(l)}=H_l \times W_l}$, $H_l \leqslant H_X$, $W_l \leqslant W_X$, полученные, например, из скрытых связей слоев нейросетевого кодировщика $Enc(\cdot)$.

1.2 Факторный анализатор с аддитивным и импульсным шумами

В задаче классификации изображений основной трудностью является выявление и обобщение глобальных структурных особенностей снимка, присущих каждому из разделяемых классов. Поэтому при выборе модели для

Рисунок 1.1 — Концепт вероятностно-информированной НС модели слияния многомасштабных признаков

информирования нейронной сети в этой задаче следует отдать предпочтение той, что способна описывать взаимодействие и реализовывать объединение не локальных (яркостных или межпиксельных), а глобальных характеристик изображения. При этом в условиях малых наборов эта модель должна быть устойчива к появлению различных видов помех.

Модель факторного анализатора получила широкое распространение для решения задачи слияния признаков с учетом влияния помех [75]. В этой вероятностной модели выходное наблюдение $Z \in \mathbb{R}^{K_Z}$ является линейной функцией матрицы факторов Θ, составленную из векторов объединяемых признаков $\theta_i \in \mathbb{R}^{K_\theta}, i = 1 \dots M_0$:

$$Z = G(\Theta) = A \cdot \Theta + \xi_0 = A \cdot (\Theta + A^{-1} \cdot \xi_0) = A \cdot (\Theta + \xi), \qquad (1.2)$$

где M_0 – число факторов модели, а K_θ и $K_Z \in \mathbb{N}$ – размерности фактора и результата слияния соответственно, $A \in \mathbb{R}^{K_Z \times K_\theta}$ – матрица нагрузок, а ξ_0 (и ξ) – ненаблюдаемое шумовое слагаемое, которое выражает аддитивную ошибку, то есть $ns(x) = x + \xi$. Обычно ξ – непрерывная случайная величина (с.в.) с нулевым средним, которая распределена по нормальному закону с диагональной

матрицей ковариаций, заданная на стандартном вероятностном пространстве $(\Omega, \mathcal{F}, \mathbb{P})$, где $\Omega = \mathbb{R}^{K_Z}$, $\mathcal{F}$– борелевская сигма-алгебра событий и $\mathbb{P}$ – вероятностная мера. При таком выборе шумового слагаемого у параметров факторного анализатора существуют простые аналитические оценки [154].

Для обработки не-Гауссовских искажений были разработаны модели-обобщения, такие как смеси факторных анализаторов, использующие аппроксимации как нормальными [117; 155], так и асимметрично распределенными компонентами [156]. Оценка параметров факторного анализатора аналитическими методами требует предположения о непрерывности распределения факторов и шумовой компоненты, однако не для всех видов искажений, например, импульсного шума [157; 158], такое представление корректно. Импульсный шум не является непрерывным, и поэтому моделирование таких искажений в факторных анализаторах в литературе не встречается. Такой шум может быть представлен, например, как произведение нормально распределенной с.в. на биномиальную с.в. с вероятностью успеха d_{prob} [159].

Предположим, что импульсная помеха полностью искажает полезную информацию в сигнале. Тогда вклад в наблюдаемое значение Z от искаженных элементов не должен нести никакой полезной информации. Предположим, что все искаженные элементы могут быть приравнены к постоянному значению y_0^*. Факторную модель с такой упрощенной импульсной помехой можно записать в следующем виде:

$$\begin{aligned} Z &= A \cdot (\eta \cdot (\Theta + \xi_0) + y_0^* \cdot (I - \eta)) = \\ &= \eta \cdot (A \cdot \Theta + \xi_0) + y_0 \cdot (I - \eta), \end{aligned} \tag{1.3}$$

где η – матрица размера $K_Z \times K_Z$, на диагоналях которой расположены независимые одинаково распределенные биномиальные с.в. с вероятностью успеха d_{prob}, $\eta_i \sim Bi(d_{prob})$, I – единичная матрица, а y_0 – некоторое постоянное значение. Согласно формуле (1.3), каждый элемент или с вероятностью d_{prob} искажен импульсной помехой и не несет полезной информации, или представляется выражением $(\Theta + \xi_0)$, обозначающим наблюдаемый сигнал, искаженный нормально распределенной аддитивной помехой. Рассмотрим аналитические свойства факторного анализатора с аддитивным и импульсным шумами.

Теорема 1. *Система случайных величин, составляющих факторную модель (1.3) с импульсным и аддитивным шумами, идентифицируема.*

Доказательство. Рассмотрим с.в. $\gamma = A \cdot \Theta + \xi_0$ из формулы (1.3). Она распределена по нормальному закону, поскольку $A \cdot \Theta$ – это неслучайные факторы модели, а $\xi_0 \sim N(0,\Sigma)$. Тогда в формуле (1.3) Z с вероятностью d_{prob} получено из γ, а с вероятностью $(1 - d_{prob})$ принимает значение y_0. Тогда случайная величина Z является номально-биномиальной смесью [160] – конечной смесью распределений непрерывной с.в. γ и дискретной y_0. Функции распределения этих с.в. не могут быть линейно выражены друг из друга, соответственно, составленная из них система будет линейно независимой. Тогда по необходимому и достаточному условию идентифицируемости [161], смесь Z будет идентифицируема. □

Теорема 2. *Пусть факторы* θ_i, $i = 1 \dots M_0$ *центрированы,* $\xi_0 \sim N(0,\Sigma)$, $y_0 = 0$, *а вероятность* d_{prob} *известна. Тогда достаточное условие однозначной определенности матриц* A, Σ *состоит в том, что размерности* Z *и* Θ *связаны соотношением* $K_Z > 2 \cdot K_\theta + 1$.

Доказательство. Пусть A, Σ определены однозначно. Рассмотрим тогда матрицу ковариаций величины Z:

$$\Psi = \mathbb{E}(Z - \mathbb{E}Z)^2 = \mathbb{E}(d_{prob} \cdot (A \cdot \Theta + \xi_0) + \\ + (1 - d_{prob}) \cdot y_0)^2 = d_{prob}^2(\cdot A \cdot \mathbb{E}(\Theta \cdot \Theta^T) \cdot A^T + \Sigma). \tag{1.4}$$

Предположим, что Θ – случайная величина. Поскольку по условию Θ центрирована, $\Psi = d_{prob}^2 \cdot (A \cdot A^T + \Sigma)$. Это выражение задает квадратичную форму. Если d_{prob} известно, то тогда достаточными условиями однозначной определенности матриц A, Σ являются условия теоремы Андерсона-Рубина [75], а именно $K_Z > 2 \cdot K_\theta + 1$. Следуют они из теоремы Витта о разложении конечномерного векторного пространства, снабженного невырожденной квадратичной формой [162]. Векторным пространством, задаваемым Ψ, является пространство, строящееся на множестве векторов-строк матрицы A. Согласно теореме, размерность его равняется $2 \cdot K_\theta + 1$ (размерность матрицы W в декомпозиции Витта равна 1, поскольку существует только один тривиальный сингулярный вектор для квадратичной формы A). Соответственно, число строк K_Z матрицы A должно превосходить это значение. □

Теперь рассмотрим свойства оценок параметров факторного анализатора с импульсным и аддитивным шумами.

Теорема 3. *Пусть* $\mathbb{E}(\Theta) = 0$, *случайные величины* η, Θ, ξ_0 *независимы, а значения* y_0, d_{prob} *известны. Тогда:*

- *если* $\xi_0 \sim N(y_0, \Sigma_{\xi_0})$, *где* $\Sigma_{\xi_0} = \sigma^2 \cdot I$, *оценки метода наименьших квадратов (МНК-оценки) матрицы* A *являются несмещенными и состоятельными;*
- *если* $\xi_0 \sim N(0, \Sigma)$, *где* $\Sigma_{\xi_0} = \sigma^2 \cdot I$, *то МНК-оценка* A *является ее оценкой максимального правдоподобия (ОМП-оценкой);*
- *если* $\xi_0 \sim N(0, \Sigma)$, *где* $\Sigma_{\xi_0} = \sigma^2 \cdot I$, *и оценки параметров* A, Σ *получены при минимизации кросс-энтропии, то эти оценки являются несмещенными и состоятельными.*

Доказательство. Докажем первое утверждение. Рассмотрим величину $Z = \eta \cdot (A \cdot \Theta + \xi_0) + y_0 \cdot (I - \eta)$. Перегруппируем слагаемые:

$$Z = A \cdot \eta \cdot \Theta + \eta \cdot \xi_0 + y_0 \cdot (I - \eta).$$

Обозначим $\Phi = \eta \cdot \Theta$, $\Psi = \eta \cdot \xi_0 - y_0 \cdot \eta$ и $\hat{Z} = Z - y_0 \cdot I$ и получим выражение $\hat{Z} = A \cdot \Phi + \Psi$. $\hat{Z}$ – факторный анализатор со стохастическими факторами $\Phi = \eta \cdot \Theta$. При этом $\mathbb{E}(\Phi) = 0$, так как η и Θ независимые случайные величины и $\mathbb{E}(\eta \cdot \Theta) = \mathbb{E}(\eta) \cdot \mathbb{E}(\Theta)$. Аналогично $\mathbb{E}(\Psi) = \mathbb{E}(\eta \cdot (\xi_0 - y_0)) = \mathbb{E}(\eta) \cdot \mathbb{E}(\xi_0 - y_0) = 0$.

Обозначим матрицу ковариаций Φ как Σ_Φ:

$$\Sigma_\Psi = \mathbb{E}\Psi^2 = \mathbb{E}(\eta^2 \cdot (\xi_0 - y_0)^2) = (d_{prob} \cdot (1 - d_{prob}) + d_{prob}^2) \cdot \sigma^2 \cdot I = d_{prob} \cdot \sigma^2 \cdot I. \quad (1.5)$$

Матрицы Φ и Ψ некоррелированы, что можно проверить непосредственно, вычислив ковариацию Φ и Ψ:

$$cov(\Phi, \Psi) = \mathbb{E}(\Phi \cdot \Psi) - \mathbb{E}(\Phi) \cdot \mathbb{E}(\Psi) =$$
$$= \mathbb{E}(\eta^2 \cdot \Theta \cdot (\xi_0 - y_0)) = \mathbb{E}(\eta^2) \cdot \mathbb{E}(\Theta) \cdot \mathbb{E}(\xi_0 - y_0) = 0.$$

МНК-оценка матрицы A будет иметь вид $\hat{A}^T = (\Phi \cdot \Phi^T)^{-1} \Phi \cdot \hat{Z} = A^T + (\Phi \cdot \Phi^T)^{-1} \Phi \cdot \Psi$. Эта оценка является линейной по параметрам, несмещенной и состоятельной. Докажем несмещенность:

$$\mathbb{E}(\hat{A}^T) = \mathbb{E}(A^T) + \mathbb{E}((\Phi \cdot \Phi^T)^{-1} \Phi \cdot \Psi) = A^T + \mathbb{E}((\Phi \cdot \Phi^T)^{-1} \Phi) \cdot \mathbb{E}(\Psi) = A^T.$$

Состоятельность (N - число наблюдений выборки):

$$\hat{A}^T = A^T + (\Phi \cdot \Phi^T)^{-1} \Phi \cdot \Psi = A^T + (\Phi \cdot \Phi^T)^{-1} \cdot (N \cdot N^{-1}) \Phi \cdot \Psi =$$
$$= A^T + (N^{-1} \cdot \Phi \cdot \Phi^T)^{-1} \cdot N^{-1} \Phi \cdot \Psi \to A^T.$$

Докажем второе утверждение теоремы. Согласно теореме 1, выражение (1.3) есть смесь распределений с.в. $\gamma \sim N(A \cdot \Theta, \Sigma)$ и вырожденной дискретной с.в. γ_{y_0}, принимающей значение y_0, то есть $P(\gamma_{y_0} = y_0) = 1$ и $P(\gamma_{y_0} \neq y_0) = 0$. Функция распределения этой смеси имеет вид:

$$H(y) = d_{prob} \cdot F_\gamma(y) + (1 - d_{prob}) \cdot I(y_0), \tag{1.6}$$

где $F_\gamma(y)$ – функция распределения γ, а $I(y)$ – индикатор. Рассмотрим логарифм правдоподобия смеси:

$$ln(L(y,A,\Sigma)) = \sum_i ln(d_{prob} \cdot f_\gamma(y_i) + (1 - d_{prob}) \cdot P(\gamma_{y_0} = y_i)), \tag{1.7}$$

где $f_\gamma(y_i) = \frac{exp^{-0.5 \cdot (y_i - A \cdot \theta) \cdot \Sigma^{-1} \cdot (y_i - A \cdot \theta)^T}}{(2 \cdot \pi)^{\frac{K_Z}{2}} \cdot det(\Sigma)}$ – плотность γ. Поскольку d_{prob}, y_0 известны, то $ln(L(y,A,\Sigma)) \to max$ тогда и только тогда, когда $\sum_i ln(f_\gamma(y_i)) \to max$. Натуральный логарифм – возрастающая функция. Она стремится к максимуму, когда $\sum_i f_\gamma(y_i) \to max$. В свою очередь, эта сумма стремится к максимуму, когда $\sum_i (y_i - A \cdot \theta) \cdot (y_i - A \cdot \theta)^T \to min$. Отсюда следует эквивалентность МНК-оценки и ОМП-оценки для матрицы A в предложенной факторной модели.

Наконец, докажем третье утверждение. Кросс-энтропия целевого и предсказываемого распределений $p(x)$ и $q(x)$ имеет вид:

$$H(p,q) = D_{KL}(p|q) - p(x) \sum ln(q(x)), \tag{1.8}$$

где $D_{KL}(p|q)$ – расстояние Кульбака-Лейблера, минимизация которого эквивалентна максимизации отношения правдоподобия [163]. Поэтому для факторного анализатора с аддитивным и импульсным шумами оценки A, полученные минимизацией кросс-энтропии, асимптотически являются ОМП, что означает выполнение для них свойств несмещенности и состоятельности согласно двум доказанным выше утверждениям теоремы. □

Теоремы 1 и 2 доказывают, что параметры факторного анализатора, заданного формулой (1.3), могут быть определены однозначно. Теорема 3 демонстрирует, что оценки параметров предложенной факторной модели, полученные, например, в ходе оптимизации кросс-энтропии нейросетевым классификатором,не накапливают систематические искажения и стремятся к своим истинным значениям. Эти результаты обосновывают выбор предложенной модели факторного анализатора для информирования сети в условиях обработки малых наборов.

1.3 Архитектура FtFNN

Модель факторного анализатора не накладывает явных ограничений на значение целевой переменной Z, в отличие, например, от системы дифференциальных уравнений. При этом для повышения точности решения задачи классификации эта модель должна реализовывать взаимодействие глобальных признаков изображения. Поэтому информирование моделью факторного анализатора должно быть реализовано на уровне архитектуры сети в рамках информированной процедуры слияния признаков.

Для этого в работе предлагается новая архитектура Factor Fusion Neural Network (FtFNN) для классификации изображений. Ее схема представлена на рисунке 1.2. Большинство стандартных сетей для классификации изображений состоит из двух частей: кодировщика признаков и классификатора, формирующего метку класса. FtFNN сохраняет эту структуру, при этом в качестве классификатора в этой архитектуре выступает блок $G(\cdot)$, выполняющий слияние M_0 многомасштабных признаков, получаемых из разных слоев сети-кодировщика. Классификатор $G(\cdot)$ информирован моделью факторного анализатора с импульсной и аддитивной помехами: искомые вероятности классов изображения $p_k, k = \overline{1,K}$ считаются наблюдаемыми величинами Z, а разномасштабные глобальные признаки изображения, в качестве которых берутся активации промежуточных слоев кодировщика – детерминированными факторами θ_i. Информирование реализовано на уровне архитектуры сети, что означает, что вычисление вероятностей классов в $G(\cdot)$ включает три этапа, на которые разбивается алгоритм вычисления таких вероятностей в математической модели факторного анализатора с импульсно-аддитивным шумом:

- **Отображение в пространство факторов.** Входные признаки θ_i проецируются с помощью процедуры P_F в пространство факторов R_F, притом P_F приводит факторы из разномасштабных слоев кодировщика к одной размерности (см. раздел 1.3.1).
- **Учет шумовой составляющей.** После отображения в пространство факторов к $P_F(\theta_i)$ добавляется сложная шумовая компонента ξ для учета импульсной и аддитивной помех (см. раздел 1.3.2).
- **Умножение на матрицу нагрузок.** Умножение на матрицу нагрузок A реализовано с помощью полносвязного слоя. Полученные значения

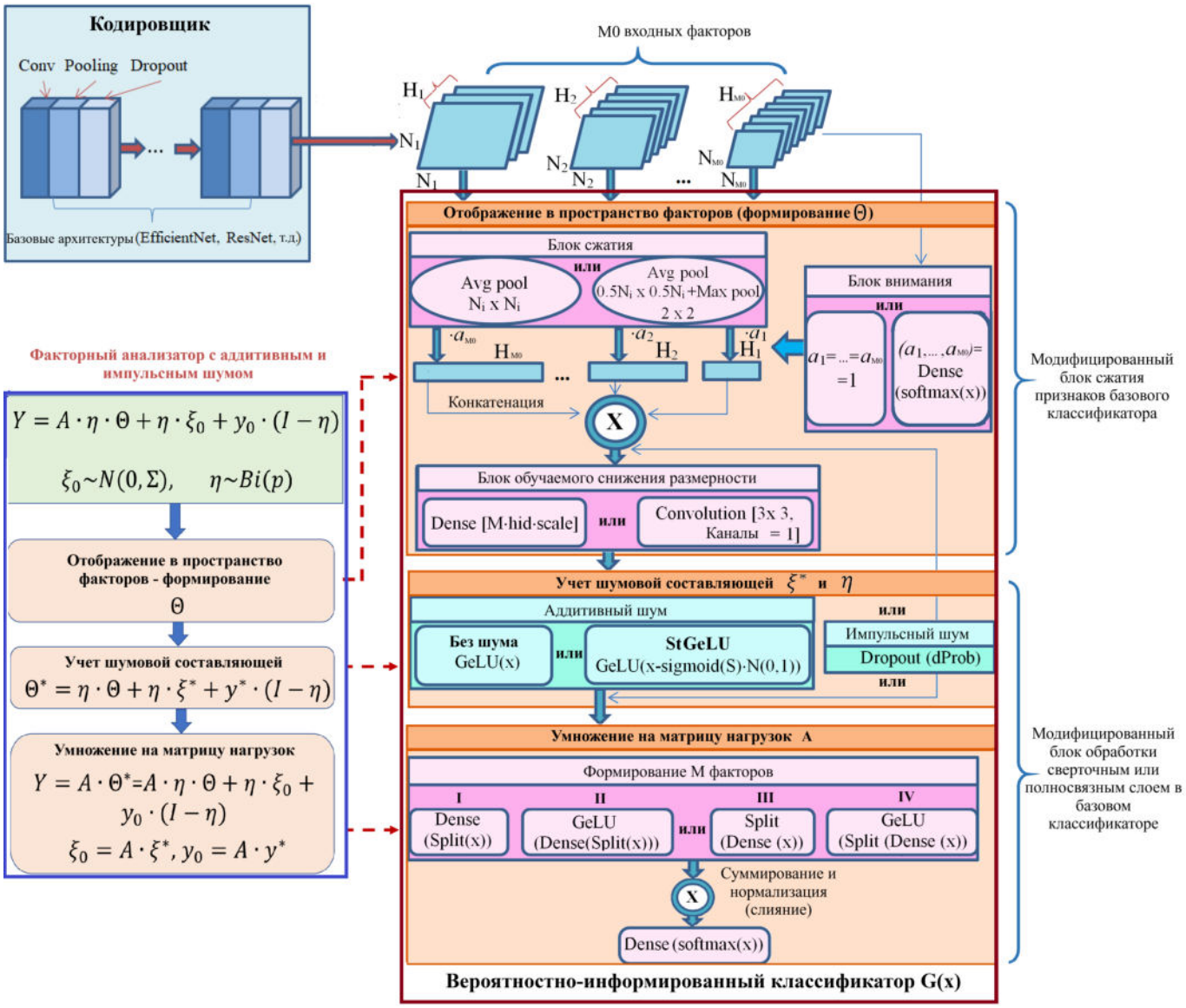

Рисунок 1.2 — Архитектура FtFNN

суммируются, нормализуются после чего принимаются равными вектору вероятностей классов $P = \{p_k\}_{k=1}^{K}$ (см. раздел 1.3.3).

Чтобы удовлетворить условиям теоремы о несмещенности и состоятельности оценок факторного анализатора (см. раздел 1.2), факторы θ_i были нормализованы, а аддитивная составляющая шумовой компоненты ξ являлась центрированной нормально распределенной случайной величиной. Все три шага вычисления p_k в архитектуре FtFNN настраиваются с помощью гиперпараметров, подробно описанных в разделах 1.3.1–1.3.3.

1.3.1 Отображение в пространство факторов

В $G(\cdot)$ первым этапом вычисления вероятностей p_k является отображение $P_F(\cdot))$ входных данных в пространство факторов: факторы приводятся к общей размерности для формирования из них матрицы Θ. $P_F(\cdot)$ реализовано блоком сжатия $Pooling(\cdot)$, представляющим собой композицию слоев пулинга, и блоком обучаемого снижения размерности, включающим вычисление весов факторов и их пост-обработку с помощью полносвязных или сверточных слоев.

На вход блоку сжатия $Pooling(\cdot)$ подаются представления входного изображения, сформированные во внутренних слоях кодировщика $\theta_i, i = \overline{1, M_0}, \theta_i \in \mathbb{R}^{N_i \times N_i \times H_i}$, где N_i – высота и ширина θ_i, а H_i – количество каналов. В общем случае размерности могут быть упорядочены: $H_1 \leqslant \ldots \leqslant H_{M_0}$, $N_{M_0} \leqslant \ldots \leqslant N_1$,. Размерность признаков θ_i уменьшается до $1 \times H_i$ за счет пулинга в среднем с полем $N_i \times N_i$ или его комбинации с макс-пулингом с полем 2×2:

$$Pooling(\theta_i) = \begin{cases} avgpool(\theta_i, N_i \times N_i), \\ maxpool(avgpool(\theta_i, \frac{N_i}{2} \times \frac{N_i}{2}), 2 \times 2). \end{cases} \tag{1.9}$$

Применение пулинга в среднем уменьшает дисперсию признаков, а макс-пулинг при небольшом размере поля позволяет выделить важные для различения объектов признаки данных (при большом поле наоборот наблюдается потеря значимой информации). Согласно формуле (1.9) для M_0 входных факторов θ_i существует 2^{M_0} способов уменьшить их размерность. Выбор способа является гиперпараметром модели.

В блоке обучаемого снижения размерности обработка признаков разделена на два этапа. На первом этапе вычисляются веса $a_i, i = \overline{1, M_0}$, признаков $Pooling(\theta_i)$. Эти веса могут как быть равными друг другу, так и настраиваться с помощью специального блока внимания, получающего на вход θ_{M_0}: $\hat{a} = (a_1, \ldots, a_{M_0}) = softmax(Dense(\theta_{M_0}))$ и $Att_{weight}(Pooling(\theta_i)) = \hat{a} \odot Pooling(\theta_i)$, $Att_{weight}(Pooling(\theta_i)) \in \mathbb{R}^{1 \times H}$, $H = \overline{H_1, H_{M_0}}$, а $\odot$ – операция поэлементного произведения.

Затем взвешенные факторы обрабатываются полносвязным или сверточным (далее обозначается символом $Lin|Conv$) слоем с размером ядра 3×3, который и далее уменьшает их размерность: общая формула проецирования в пространство факторов имеет вид:

$$P_F = Lin|Conv \circ Att_{weight} \circ Pooling. \tag{1.10}$$

Полносвязный слой уменьшает размерность $Att_{weight}(Pooling(\theta_i))$ до величины $1 \times M \cdot hid \cdot scale$, где M число факторов в пространстве скрытых признаков ($M \leqslant M_0$), hid – размерность фактора, а $scale \in \mathbb{N}$ – масштаб фактора, то есть количество элементов вектора соответствующих каждой позиции hid.

Для обработки сверточным слоем признаки $Att_{weight}(Pooling(\theta_i))$ преобразуется к тензору размерности $2^{I_{Max}/2} \times 2^{I_{Max}/2} \times H \cdot 2^{-I_{Max}}$, где I_{Max} – максимальный показатель степени при котором H нацело делится на $2^{-I_{Max}}$. Сформированный тензор имеет вид изображения с числом каналов $H \cdot 2^{-I_{Max}}$, которое с помощью свертки уменьшается до 1. Если $2^{I_{Max}/2}$ мало (например, равняется 2 или 4), ширина изображения принимается равной $2^{-I_{Max}}$, а его высота рассматривается как гиперпараметр.

1.3.2 Учет шумовой составляющей

На втором шаге вычисления p_k в $G(\cdot)$ к преобразованным факторам добавляется шумовая составляющая. Учет аддитивной компоненты реализован специально разработанной функцией активации *StochasticGeLU* (*StGeLU*). Эта функция является модификацией $GeLU(x) = x \cdot \Phi(x)$ [164], где $\Phi(x)$ – функция распределения стандартного нормального закона. *StGeLU*, как и *GeLU*, уменьшает интенсивность отклика от отрицательных элементов вектора x, но при этом учитывает искажения этих элементов за счет вычитания из сигнала шумовой составляющей:

$$StGeLU(x) = GeLU(x - Sigmoid(\sigma) \cdot \varepsilon), \tag{1.11}$$

где $\varepsilon \sim N(0,1)$, а обучаемый параметр σ – интенсивность шумовой компоненты, не превосходящая интенсивности полезного сигнала. *GeLU* используется вместо *StGeLU* если $\sigma = 0$.

StGeLU обладает некоторым сходством с функцией активации *ProbAct* [165], однако в отличие от нее учитывает шумовую составляющую до уменьшения интенсивности отрицательных элементов сигнала, что ведет к меньшей потере полезной информации.

В соответствии с предположениями модели, искаженные импульсным шумом элементы сигнала не содержат полезной информации и должны приниматься равными некоторому константному значению $\bar{y}_0$, например, среднему входных факторов. Для нормализованных факторов обработка импульсной составляющей может быть реализована слоем дропаута с вероятностью отбрасывания элементов d_{prob}.

1.3.3 Умножение на матрицу нагрузок

На третьем шаге значения M факторов $P_F^{\xi}(\Theta) = \eta(P_F(\Theta) + \xi) + y_0(I - \eta)$, полученные на двух предыдущих этапах, умножаются на матрицу нагрузок A, после чего выполняется формирование вектора вероятностей классов $P = \{p_k\}_{k=1}^{K}$. Он вычисляется как сумма преобразованных факторов $A^{(j)}(P_F^{\xi}(\Theta))$, где $A^{(j)}(x)$ означает умножение на матрицу A одним из четырех способов, включающих этап разделения вектора признаков $P_F^{\xi}(\Theta)$ на M равных частей (факторов):

$$A^{(I)}(P_F^{\xi}(\Theta)) = \sum_{i=1}^{M} Dense_i(Split(P_F^{\xi}(\Theta), M)_i),$$

$$A^{(II)}(P_F^{\xi}(\Theta)) = \sum_{i=1}^{M} GeLU(Dense_i(Split(P_F^{\xi}(\Theta), M)_i)),$$

$$A^{(III)}(P_F^{\xi}(\Theta)) = \sum_{i=1}^{M} Split(GeLU(Dense(P_F^{\xi}(\Theta))), M)_i,$$

$$A^{(IV)}(P_F^{\xi}(\Theta)) = \sum_{i=1}^{M} Split(Dense(P_F^{\xi}(\Theta)), M)_i,$$

где $Split(P_F^{\xi}(\Theta), M)_i$ – i-я часть разделяемого вектора $P_F^{\xi}(\Theta)$, а $Dense_i(\cdot)$ – его преобразование одним из M одинаковых полносвязных слоев. Для выполнения вероятностных свойств вектора P, сумма $A^{(j)}(P_F^{\xi}(\Theta))$ нормируется процедурой $Norm(\cdot)$. Тогда полный алгоритм вычисления P может быть описан формулой:

$$P = softmax(Dense(Norm(A^{(j)}((P_F^{\xi}(\Theta)))))). \tag{1.12}$$

1.3.4 Оценка вычислительной сложности информированного классификатора FtFNN

Сравним вычислительную сложность классификатора FtFNN $G(\cdot)$ со стандартными нейросетевыми классификаторами без информирования: на основе полносвязных или сверточных слоев. В первом случае простейший вариант классификатора, реализованный, например, в архитектурах EfficientNet [135] или XCeption [136] – последовательность из слоя пулинга в среднем и полносвязного слоя, принимающая на вход только окончательный результат обработки входного изображения энкодером – тензор размерности $N_{M_0} \times N_{M_0} \times H_{M_0}$. Вычислительная сложность такого классификатора складывается из сложности поканального суммирования признаков в слое пулинга и сложности матричного умножения в полносвязном слое:

$$T_{base}^{(1)} = \underbrace{H_{M_0} \cdot K}_{\text{Сложность полносвязного слоя}} + \underbrace{N_{M_0}^2 \cdot H_{M_0}}_{\text{Сложность пулинга}} . \tag{1.13}$$

Классификатор на основе сверточных слоев реализован, например, в архитектуре MobileNetV3 [166]. Он имеет вид последовательности из слоя пулинга и двух слоев двумерной свертки, где первая увеличивает число каналов до величины $chnl$, а вторая уменьшает его до числа классов K. Вычислительная сложность такого блока описывается формулой:

$$T_{base}^{(2)} = \underbrace{H_{M_0} \cdot (N_{M_0} - S_1 + 1)^2 \cdot chnl \cdot S_1^2}_{\text{Сложность первой свертки}} + \tag{1.14}$$

$$+ \underbrace{K \cdot chnl \cdot (N_{M_0} - S_2 + 1)^2 \cdot S_2^2}_{\text{Сложность второй свертки}} + \underbrace{N_{M_0}^2 \cdot H_{M_0}}_{\text{Сложность пулинга}} ,$$

где S_1, S_2 – размеры поля первого и второго сверточных слоев. Сложность сверточного слоя зависит от сложности матричного умножения матрицы поля и количества сдвигов, необходимых для покрытия этой матрицей всего изображения.

Вычислительная сложность классификатора FtFNN T_{FtFNN} складывается из оценок сложности трех этапов его работы, описанных в разделах 1.3.1–1.3.3. Для первого этапа – отображения в пространство факторов – вычислительная сложность $T_{FtFNN}^{(1)}$ может быть оценена двумя способами

в зависимости от реализации блока обучаемого снижения размерности.

$$T^{(1)}_{FtFNN} = \begin{cases} H \cdot F^{(1)}_{scale} + Pool, \text{ для полносвязного слоя,} \\ T^{(1)}_{FtFNN} = 9 \cdot \frac{H}{(F^{(2)}_{scale})^2} \cdot (F^{(2)}_{scale} - 2)^2 + Pool, \text{ для сверточного слоя,} \end{cases}$$

где $Pool$ – число операций, выполняемых в блоке сжатия, $H = \sum_{i=1}^{3} H_i$, $F^{(1)}_{scale} = M \cdot hid \cdot scale$ – размерность вектора признаков после обработки его полносвязным слоем, а $F^{(2)}_{scale} = \sqrt{2^{I_{Max}/2}}$ – сверточным. Величина $Pool$ также оценивается двумя способами:

$$Pool = \begin{cases} N_i^2 \cdot H_i, \text{ для пулинга в среднем,} \\ \left(\left(\frac{N_i}{2}\right)^2 + 2 \cdot 2\right) \cdot H_i. \text{ для комбинации двух видов пулинга.} \end{cases}$$

Сложность учета шумовой составляющей (второй этап) $T^{(2)}_{FtFNN}$ равняется или $F^{(1)}_{scale}$, или $F^{(2)}_{scale}$ в зависимости от использования функции активации StGeLU или GeLU. Наконец, для этапа умножения на матрицу нагрузок (третий этап) справедливы две оценки вычислительной сложности, зависящие от использования блоков (I) и (II) или (III) и (IV), отмеченных на рисунке 1.2 (F_{scale} обозначает $F^{(1)}_{scale}$ или $F^{(2)}_{scale}$ в зависимости от способа реализации учета шумовой компоненты):

$$T^{(3)}_{FtFNN} = \begin{cases} F_{scale} \cdot hid \cdot scale + hid \cdot scale \cdot K, \text{ для (I), (II) блоков,} \\ M \cdot hid \cdot scale \cdot F_{scale} + hid \cdot scale \cdot K, \text{ для (III), (IV) блоков.} \end{cases}$$

Теорема 4. *Пусть число факторов в скрытом пространстве $M \in \mathbb{N}$, параметры $hid = K$, $M \cdot scale \cdot K \in \mathbb{N}$ и веса факторов a_i являются фиксированными параметрами, а для уменьшения размерности выходного слоя кодировщика $N_{M_0} \times N_{M_0} \times H_{M_0}$ ($Pool_{sub} = Pool - N^2_{M_0} \cdot H_{M_0}$) используется пулинг в среднем. Тогда число операций, выполняемых в классификаторе FtFNN, меньше, чем в базовом классификаторе на основе полносвязного слоя (формула (1.13)) если:*

1. $M < \frac{H_{M_0} - \frac{Pool_{sub}}{K}}{(H - (1 + K \cdot scale)) \cdot scale}$, *когда первый этап реализован полносвязным слоем, а третий – блоками (I) или (II);*
2. $M < \frac{\sqrt{(H+1)^2 + 4 \cdot (H_{M_0} - \frac{Pool_{sub}}{K}) \cdot K} - (H+1)}{2 \cdot K \cdot scale}$, *когда первый этап реализован полносвязным слоем, а третий – блоками (III) или (IV);*

3. $scale < \frac{H_{M_0} \cdot K - Pool_{sub} - (\frac{9 \cdot (H) \cdot (F^{(2)}_{scale} - 2)^2}{(F^{(2)}_{scale})^2} + (F^{(2)}_{scale})^2)}{((F^{(2)}_{scale})^2 + K) \cdot K}$ *и* $K > 9 \cdot \frac{(F^{(2)}_{scale} - 2)^2}{(F^{(2)}_{scale})^2}$, *когда первый этап реализован сверточным слоем, а третий – блоками (I) или (II);*
4. $scale < \frac{H_{M_0} \cdot K - Pool_{sub} - (\frac{9 \cdot H \cdot (F^{(2)}_{scale} - 2)^2}{(F^{(2)}_{scale})^2} + (F^{(2)}_{scale})^2)}{(M \cdot (F^{(2)}_{scale})^2 + K) \cdot K}$ *и* $K > 9 \cdot \frac{(F^{(2)}_{scale} - 2)^2}{(F^{(2)}_{scale})^2}$, *когда первый этап реализован сверточным слоем, а третий – блоками (III) или (IV).*

Также число операций, выполняемых в классификаторе FtFNN, меньше, чем в базовом классификаторе на основе сверточных слоев (формула (1.14)) если:

1. $M < \frac{T^{(2)}_{base} - Pool_{sub} - scale \cdot K^2}{(H+1+K \cdot scale) \cdot K \cdot scale}$, *когда первый этап реализован полносвязным слоем, а третий – блоками блоками (I) или (II);*
2. $M < \frac{\sqrt{(H+1)^2 + 4(T^{(2)}_{base} - Pool_{sub} - scale \cdot K^2)} - (H+1)}{2 \cdot K \cdot scale}$, *когда первый этап реализован полносвязным слоем, а третий – блоками (III) или (IV);*
3. $scale < \frac{T^{(2)}_{base} - Pool_{sub} - (\frac{9 \cdot H \cdot (F^{(2)}_{scale} - 2)^2}{(F^{(2)}_{scale})^2} + (F^{(2)}_{scale})^2)}{((F^{(2)}_{scale})^2 + K) \cdot K}$, *когда первый этап реализован сверточным слоем, а третий – блоками блоками (I) или (II);*
4. $scale < \frac{T^{(2)}_{base} - Pool_{sub} - (\frac{9 \cdot H \cdot (F^{(2)}_{scale} - 2)^2}{(F^{(2)}_{scale})^2} + (F^{(2)}_{scale})^2)}{(M \cdot (F^{(2)}_{scale})^2 + K) \cdot K}$, *когда первый этап реализован сверточным слоем, а третий – блоками (III) или (IV).*

Доказательство. Обозначим I индикатор реализации умножения на матрицу нагрузок (третий этап) блоками (I), (II). Тогда в случае, когда используются блоки (III), (IV) $I = 0$, иначе – $I = 1$. Рассмотрим вначале случай, когда первый этап реализован полносвязным слоем:

$$T_{FtFNN} = (H + 1 + (M \cdot I + (1 - I)) \cdot K \cdot scale) \cdot F^{(1)}_{scale} + Pool + scale \cdot K^2. \quad (1.15)$$

Параметры $H, K, F^{(2)}_{scale} = 2^{I_{Max}/2}$ задаются конфигурацией кодировщика и потому не настраиваются. Параметры M и $scale$, напротив, изменяются. Значит, можно определить их значения, при которых классификатор FtFNN будет более простым по количеству выполняемых операций в сравнении с базовым классификатором.

Рассмотрим вначале случай, когда базовый классификатор реализован полносвязным слоем. Предположим, что в FtFNN блок обучаемого снижения размерности реализован полносвязным слоем. Приравняем тогда $F^{(1)}_{scale}$ значение

$M \cdot K \cdot scale$ и сравним $\sum_i^3 T_{FtFNN}^{(i)}$ с вычислительной сложностью полносвязного классификатора $H_{M_0} \cdot K + N_{M_0}^2 \cdot H_{M_0}$. Отсюда получим выражение:

$$scale \cdot K < H_{M_0} - \frac{Pool_{sub}}{K} - (H + 1 + (M \cdot I + (1 - I)) \cdot K \cdot scale) \cdot M \cdot scale. \quad (1.16)$$

По условию теоремы $scale$, M, K строго положительны. Из этого можно получить необходимое условие, чтобы классификатор FtFNN был более вычислительно простым, чем полносвязный:

$$H_{M_0} - \frac{Pool_{sub}}{K} - (I \cdot (H - (1 + K \cdot scale)) + (1 - I)((H + 1) - (M \cdot scale) \cdot K)) \cdot M \cdot scale > 0$$

Когда $I = 1$, величина M задается линейной функцией. В случае $I = 0$, оценка M определяется из решения квадратного уравнения относительно $M \cdot scale$, дискриминант которого $\sqrt{(H+1)^2 + 4 \cdot (H_{M_0} - \frac{Pool_{sub}}{K}) \cdot K}$ положителен, если $Pool_{sub} < \frac{(H+1)^2}{4} + H_{M_0} \cdot K$:

$$M < \frac{H_{M_0} - \frac{Pool_{sub}}{K}}{(H - (1 + K \cdot scale)) \cdot scale},$$

$$M < \frac{\sqrt{(H+1)^2 + 4 \cdot (H_{M_0} - \frac{Pool_{sub}}{K}) \cdot K} - (H + 1)}{2 \cdot K \cdot scale}.$$

Рассмотрим теперь случай, когда в классификаторе FtFNN первый этап реализован сверточным слоем:

$$T_{FtFNN} = \frac{9H(F_{scale}^{(2)} - 2)^2}{(F_{scale}^{(2)})^2} + scale \cdot K^2 + (F_{scale}^{(2)})^2 \cdot (1 + (I + M \cdot (1 - I))(K \cdot scale)) + Pool.$$

Сравним выражение с $H_{M_0} \cdot K + N_{M_0}^2 \cdot H_{M_0}$. Тогда:

$$scale < \frac{H_{M_0} \cdot K - Pool_{sub} - (\frac{9 \cdot H \cdot (F_{scale}^{(2)} - 2)^2}{(F_{scale}^{(2)})^2} + (F_{scale}^{(2)})^2)}{((I + (1 - I) \cdot M) \cdot (F_{scale}^{(2)})^2 + K) \cdot K}.$$

В этом случае необходимо, чтобы одновременно было выполнено:

$$H_{M_0} \cdot K - Pool_{sub} - (\frac{9 \cdot H \cdot (F_{scale}^{(2)} - 2)^2}{(F_{scale}^{(2)})^2} + (F_{scale}^{(2)})^2) > 0.$$

Указанное выражение может быть сведено к виду:

$$H_{M_0} \cdot (K - \frac{9 \cdot (F_{scale}^{(2)} - 2)^2}{(F_{scale}^{(2)})^2}) > Pool_{sub} + \frac{9 \cdot (H - H_{M_0}) \cdot (F_{scale}^{(2)} - 2)^2}{(F_{scale}^{(2)})^2} + (F_{scale}^{(2)})^2.$$

При этом необходимым условием выполнения этого неравенства является:

$$K > 9 \cdot \frac{(F_{scale}^{(2)} - 2)^2}{(F_{scale}^{(2)})^2}. \tag{1.17}$$

Доказательство для случая, когда неинформированный классификатор реализован сверточным слоем, идентично случаю полносвязного классификатора. Пусть в классификаторе FtFNN первый этап реализован полносвязным слоем. Сравним вычислительную сложность классификатора FtFNN с вычислительной сложностью сверточного классификатора T_{base}^2. Отсюда получим условия на значений параметра M (для блоков (I), (II) или (III), (IV) третьего этапа соответственно), при которых классификатор FtFNN является более вычислительно простым, чем неинформированный сверточный:

$$M < \frac{T_{base}^{(2)} - Pool_{sub} - scale \cdot K^2}{(H + 1 + K \cdot scale) \cdot K \cdot scale},$$

$$M < \frac{\sqrt{(H+1)^2 + 4(T_{base}^{(2)} - Pool_{sub} - scale \cdot K^2)} - (H+1)}{2 \cdot K \cdot scale},$$

в случае, когда:

$$scale < \frac{T_{base}^{(2)} - Pool_{sub} + \frac{H+1}{4}}{K^2}. \tag{1.18}$$

Когда в FtFNN первый этап реализован сверточным слоем, условия вычислительной эффективности для значений параметра $scale$ имеют вид (для блоков (I), (II) или (III), (IV) третьего этапа соответственно):

$$scale < \frac{T_{base}^{(2)} - Pool_{sub} - (\frac{9 \cdot H \cdot (F_{scale}^{(2)} - 2)^2}{(F_{scale}^{(2)})^2} + (F_{scale}^{(2)})^2)}{((I + (1 - I) \cdot M) \cdot (F_{scale}^{(2)})^2 + K) \cdot K}.$$

□

Таким образом, для параметров M и $scale$, существуют диапазоны значений, при которых FtFNN является более вычислительно простой архитектурой в сравнении со стандартными полносвязными и сверточными классификаторами. Это будет продемонстрировано на практике в следующем разделе.

1.4 Классификация малых наборов изображений с помощью FtFNN

В разделе представлены результаты классификации изображений с помощью FtFNN в условиях малых обучающих наборов. Описание тестируемых данных и гиперпараметров FtFNN представлено в разделах 1.4.1 и 1.4.2. Для сравнения было выбрано несколько известных сверточных архитектур для классификации изображений: EfficientNet B0 [135], MobileNet (версии 1, 2, 3) [166—168], Xception [136], ResNet20 [169] и полностью сверточная сеть (FCN). Кодировщики этих архитектур рассматривались как варианты реализации кодировщика FtFNN. Классификаторы этих сетей были реализованы в виде полносвязных слоев за исключением MobileNetV3. Трансформерные архитектуры не рассматривались при тестировании, поскольку в случае малых по числу элементов наборов простые сверточные нейронные сети, как правило, превосходят по производительности и точности трансформеры [125].

Результаты, полученные FtFNN и архитектурами без информирования, сравнивались с использованием традиционных для оценки точности классификации метрик Top-1, Top-3 and Top-5 Accuracy [170]:

$$Top\text{-}k = \frac{\sum_{i=1}^{Total} Pr_{most}(i,k)}{Total} \cdot 100\%, \tag{1.19}$$

где $Total$ – общее количество элементов в тестовом множестве, а $Pr_{most}(i,k)$ – количество правильно предсказанных меток класса среди k наиболее вероятных. k принимает значения 1, 3 и 5. Для архитектур EfficientNet, MobileNet и Xception выполнялось предобучение на ImageNet [171], а сети ResNet20 и FCN обучались с нуля. Результаты обработки представлены в разделах 1.4.4 и 1.4.5 соответственно.

1.4.1 Описание тестируемых наборов изображений

Для тестирования FtFNN было рассмотрено несколько открытых датасетов (см. таблицу 1) по количеству содержащихся в них элементов соответ-

ствующих определению малого набора. За исключением Cifar10, все наборы содержат небольшое количество изображений: среднее количество элементов в обучающем множестве составляет 1588. Для всех наборов размер обрабатываемого изображения составлял 224×224 пикселей, за исключением Cifar10 (32×32 пикселей).

Таблица 1 — Описание тестируемых наборов изображений

Набор	Структура	Классы	Обучающий набор	Тестовый набор
Oxford flowers (OFL) [172]	1020 (OM), 6149 (TM)	102	1020 (OM)	6149 (TM)
Oxford iiit pet (OIP) [173]	3680 (OM), 3669 (TM)	37	3680 (OM)	3669 (TM)
Imagenette (INT) [174]	9469 (OM), 3925 (TM)	10	3925 (TM)	9469 (OM)
Tf flowers50 (TFL50)	3670 (OM)	5	1835 (OM 50%)	1835 (OM 50%)
Tf flowers 40 (TFL40)	3670 (OM)	5	1468 (OM 40%)	2202 (OM 60%)
Tf flowers 30 (TFL30)	3670 (OM)	5	1101 (OM 30%)	2569 (OM 70%)
UC Merced [175] 25 (UCM25)	2100 (OM)	21	525 (OM 25%)	1575 (OM 75%)
UC Merced 20 (UCM20)	2100 (OM)	21	420 (OM 20%)	1680 (OM 80%)
UC Merced 15 (UCM15)	2100 (OM)	21	315 (OM 15%)	1785 (OM 85%)
UC Merced 10 (UCM10)	2100 (OM)	21	210 (OM 10%)	17890 (OM 90%)
Cifar10 [176](CF30) 30	50000 (OM), 10000 (TM)	10	15000 (OM 30%)	35000 (OM 70%)
Cifar10 25 (CF25)	50000 (OM), 10000 (TM)	10	12500 (OM 25%)	37500 (OM 75%)
Cifar10 20 (CF20)	50000 (OM), 10000 (TM)	10	10000 (OM 20%)	40000 (OM 80%)

В столбце «Структура» таблицы 1 представлена схема разделения данных исходного набора на обучающее (OM) и тестовое (TM) множества. В столбцах «Обучающий набор» и «Тестовый набор» указаны часть исходного набора (в скобках) и количество элементов из нее, которые использовались для обучения и тестирования FtFNN. Для Tf flowers[1], Cifar10 и UC Merced [175] в исходном наборе не было специального разделения данных на обучающую и тестовую части. Поэтому первые $n\%$ (n указано через пробел) элементов набора использовались в качестве обучающего множества, а оставшиеся $(100-n)\%$ – в качестве тестового. Обучающее множество выбиралось меньшим или равным по числу элементов тестовому – при таком подходе тестовое множество гарантированно является более разнообразным, чем обучающее, и способно отражать обобщающие способности сети в условиях ограниченных наборов.

1.4.2 Гиперпараметры

Описание и диапазоны значений гиперпараметров FtFNN представлены в таблице 2. Число входных факторов M_0 во всех случаях равнялось 3 в со-

[1] http://download.tensorflow.org/example_images/flower_photos.tgz

Таблица 2 — Описание и диапазоны значений гиперпараметров FtFNN

Гиперпараметр	Описание	Значения
feature	Комбинация слоев пулинга для θ_i в блоке сжатия	1-2^{M_0}
drop_type	Обработка импульсного шума после учета аддитивного (1) или до него (0)	0; 1
dSample_type	Использование полносвязного (1) или сверточного слоев (0) для уменьшения размерности	0; 1
W_type	Использование обучаемых (0) или необучаемых (1) весов факторов	0; 1
noise	Применение GeLU (0) или StGeLU (1) для учета аддитивной помехи	0; 1
LMatr	Номер блока для умножения на матрицу нагрузок (см. рисунок 1.2)	1; 2; 3; 4
M	Число факторов модели в скрытом пространстве	1; 2; 3
dProb	Вероятность искажения элемента импульсным шумом	0.0 – 0.8
scale	Масштаб фактора	1; 2; 3
hid	Размерность фактора	K; 16; 32; 64

ответствии с количеством разных пространственных разрешений изображения, обрабатываемых внутри кодировщика.

Для обучения моделей использовалась NVidia V100. Размер батча составлял 32, число эпох обучения – 100 (для архитектур MobileNet V3, EfficientNet, ResNet20) или 200 (для остальных сетей). В качестве оптимизатора использовался Adam (он продемонстрировал лучшую сходимость в сравнении с AdamW [177] и RmsProb [178]), а в качестве функции потерь – кросс-энтропия (см. формулу (1.8)). Для всех архитектур использовалось одинаковое правило уменьшения шага обучения: начиная с величины 10^{-3} каждые 20 эпох (начиная с первой при 100 эпохах обучения, и с 80-й при 200 эпохах) уменьшалось в десять раз.

1.4.3 Общие оценки точности классификации, полученные FtFNN

В таблице 3 для каждого исследуемого датасета приведены максимальные значения Top-1 Accuracy полученные FtFNN и базовыми архитектурами (лучшие значения выделены жирным шрифтом). Согласно полученным результатам, FtFNN для каждого набора демонстрирует максимальные значения точности классификации. Прирост Top-1 Accuracy составляет 0.3-2.75% при средней точности классификации наборов базовыми архитектурами равной 89.71%. При этом наибольшие приросты точности получены на наборах OFL, UCM25, UCM20, UCM15 и UCM10, размер обучающего набора которых не превосходит 1020 примеров, а наименьшие – на подмножествах набора Cifar10, количество обучающих примеров в которых составляет от 10 до 15 тысяч. Лучшими по точности являются конфигурации с кодировщиком признаков EfficientNet и ResNet20.

Таблица 3 — Максимальные значения Top-1 Accuracy (в %), полученные FtFNN и базовыми архитектурами

Набор	Макс. Топ-1 (база)	Макс. Топ-1 (FtFNN)
OFL	85.68 (EfficientNet)	**87.70** (EfficientNet)
OIP	84.52 (EfficientNet)	**86.02**(EfficientNet)
INT	96.09 (EfficientNet)	**96.54** (EfficientNet)
TFL50	94.6 (EfficientNet)	**95.22** (EfficientNet)
TFL40	93.09 (EfficientNet)	**94.32** (EfficientNet)
TFL30	91.90 (EfficientNet)	**93.23** (EfficientNet)
UCM25	94.99 (EfficientNet)	**96.38** (EfficientNet)
UCM20	94.57 (EfficientNet)	**96.28** (EfficientNet)
UCM15	91.98(EfficientNet)	**94.73** (EfficientNet)
UCM10	88.67(EfficientNet)	**92.33** (EfficientNet)
CF30	84.07 (ResNet20)	**84.57**(ResNet20)
CF25	83.03 (ResNet20)	**83.33** (ResNet20)
CF20	83.03 (ResNet20)	**83.33**(ResNet20)

1.4.4 Результаты для FtFNN с предобученным кодировщиком

EfficientNet

В таблице 4 для девяти обучающих наборов (см. раздел 1.4.1) приведены оценки точности классификации, полученные EfficientNet B0 и построенной на основе ее кодировщика FtFNN. В таблице представлены значения Top-1, Top-3 and Top-5 Accuracy (максимальные и средние значения и их среднее квадратичное отклонение), а также число параметров для базовой сети и FtFNN (соответствующие столбцы отмечены). Сравнение максимальных показателей Top-1 Accuracy EfficientNet B0 с максимальными и средними значениями метрики, полученными FtFNN, представлено на рисунке 1.3. Знак «+» или «−» здесь и далее соответствует повышению или уменьшению точности по сравнению с результатами базовой сети.

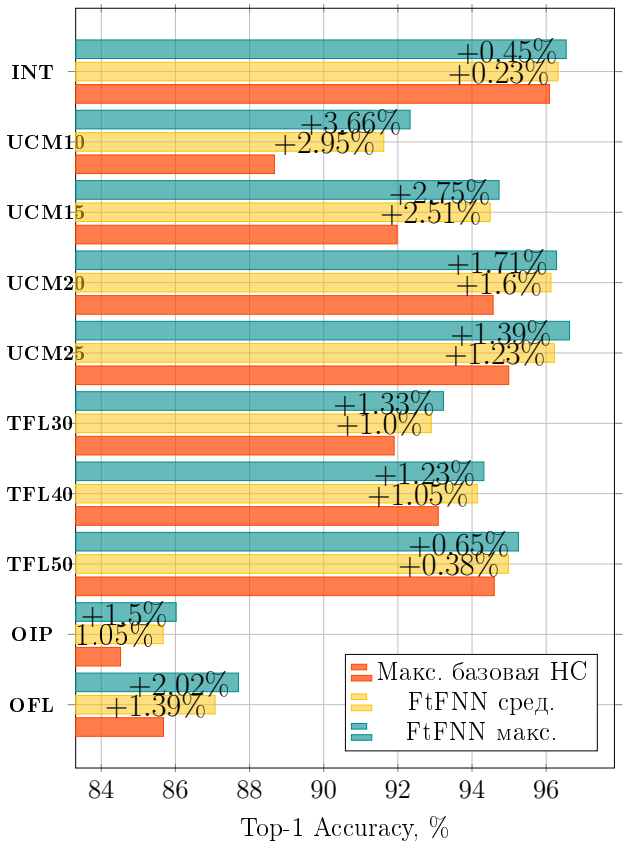

Рисунок 1.3 — Значения Top-1 Accuracy, полученные EfficientNet и FtFNN

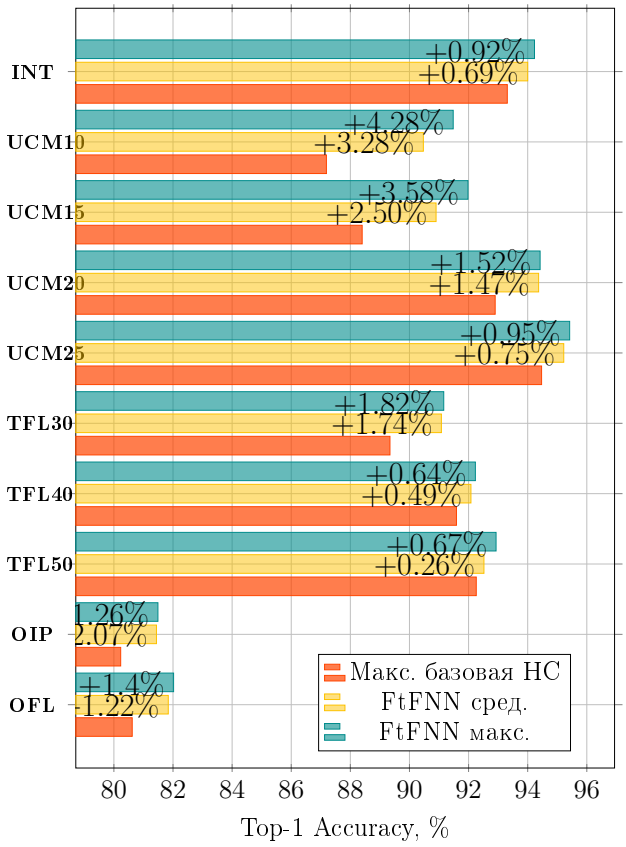

Рисунок 1.4 — Значения Top-1 Accuracy, полученные Xception и FtFNN

Таблица 4 — Значения метрики Accuracy (в %) и число параметров (млн.) для FtFNN с кодировщиком EfficientNet B0.

Набор	**Top-1 макс.**	**Top-1 макс. (FtFNN)**	**Top-1 сред.**	**Top-1 сред. (FtFNN)**	**Top-3 макс.**	**Top-3 макс. (FtFNN)**	**Top-5 макс.**	**Top-5 макс. (FtFNN)**	**Парам.**	**Парам. (FtFNN)**
OFL	85.68	**87.70**	85.66 ± 0.03	**87.07± 0.41**	93.68	**94.19**	98.65	**98.85**	6.05	**5.984**
OIP	84.52	**86.02**	83.96 ± 0.56	**85.67± 0.28**	96.15	**96.51**	98.12	**98.28**	5.966	**5.938**
TFL50	94.6	**95.22**	94.22 ± 0.54	**94.98± 0.38**	98.96	**99.45**	–	–	5.926	**5.922**
TFL40	93.09	**94.32**	92.95 ± 0.48	**94.14± 0.18**	98.85	**99.27**	–	–	5.926	**5.922**
TFL30	91.90	**93.23**	91.77 ± 0.14	**92.90 ± 0.33**	98.59	**99.02**	–	–	5.926	**5.922**
UCM25	94.99	**96.38**	94.63 ± 0.05	**96.22± 0.2**	99.30	**99.43**	99.87	**99.93**	5.946	**5.929**
UCM20	94.57	**96.28**	94.52 ± 0.26	**96.13± 0.14**	99.23	**99.24**	99.64	**99.88**	5.946	**5.929**
UCM15	91.98	**94.73**	91.50 ± 0.48	**94.49± 0.34**	98.80	**99.27**	99.64	**99.71**	5.946	**5.929**
UCM10	88.67	**92.33**	88.05 ± 0.95	**91.62± 0.62**	95.99	**97.62**	98.14	**98.80**	5.946	**5.929**
INT	96.09	**96.54**	95.33 ± 0.76	**96.32± 0.22**	99.25	**99.38**	99.76	**99.84**	5.932	**5.929**

Для всех наборов FtFNN демонстрирует более высокую точность классификации изображений в сравнении с базовой EficientNet. Для лучших показателей Top-1, Top-3 и Top-5 Accuracy прирост составляет 0.45-4.28%, 0.13-1.86%, и 0.06-0.66%, в то время как число параметров FtFNN уменьшается на 3-65 тысяч в сравнении с базовой сетью. FtFNN вычислительно проще EficientNet только на наборе OFL – число выполняемых операций уменьшается с 1.46 до 1.45 GFLOPS (здесь и далее представлены оценки, вычисленные с помощью инструмента keras_flops). Этот результат согласуется с результатами теоремы 4: для OFL верхняя граница значений параметра *scale* равняется 2.52 (а лучшие экспериментальные результаты были получены при $scale = 2$), тогда как для остальных наборов граница не превосходит 0.

Таблица 5 — Значения метрики Accuracy (в %) и число параметров (млн.) для FtFNN с кодировщиком Xception.

Набор	**Top-1 макс.**	**Top-1 макс. (FtFNN)**	**Top-1 сред.**	**Top-1 сред. (FtFNN)**	**Top-3 макс.**	**Top-3 макс. (FtFNN)**	**Top-5 макс.**	**Top-5 макс. (FtFNN)**	**Парам.**	**Парам. (FtFNN)**
OFL	80.62	**82.02**	79.19 ± 1.44	**81.84± 0.14**	88.92	**91.28**	92.66	**94.15**	21.07	**20.894**
OIP	80.23	**81.49**	79.42 ± 0.81	**81.44± 0.04**	93.37	**94.19**	96.21	**96.85**	20.937	**20.871**
TFL50	92.26	**92.93**	91.98 ± 0.28	**92.52± 0.30**	98.69	**99.34**	–	–	20.872	**20.864**
TFL40	91.59	**92.23**	91.25 ± 0.34	**92.08± 0.16**	98.81	**99.09**	–	–	20.872	**20.864**
TFL30	89.34	**91.16**	88.61 ± 1.01	**91.08 ± 0.12**	98.01	**98.36**	–	–	20.872	**20.864**
UCM25	94.47	**95.42**	94.06 ± 0.58	**95.22± 0.02**	98.92	**99.17**	99.74	**99.82**	20.905	**20.867**
UCM20	92.9	**94.42**	92.66 ± 0.24	**94.37± 0.04**	98.0	**98.19**	98.95	**99.38**	20.905	**20.867**
UCM15	88.4	**91.98**	88.18 ± 0.23	**90.90 ± 0.76**	97.19	**98.99**	98.76	**99.66**	20.905	**20.867**
UCM10	87.19	**91.47**	85.76 ± 1.26	**90.45 ± 1.13**	94.52	**96.38**	97.66	**98.09**	20.905	**20.867**
INT	93.31	**94.23**	92.06 ± 1.26	**94.00± 0.23**	98.37	**98.66**	99.46	**99.62**	20.882	**20.864**

Xception

В таблице 5 для девяти обучающих наборов (см. раздел 1.4.1) приведены оценки точности классификации, полученные Xception и построенной на ее основе FtFNN. Значения метрики Top-1 Accuracy, полученные двумя архитектурами, сравниваются на рисунке 1.4. Для всех наборов данных FtFNN демонстрирует более высокую точность классификации изображений в сравнении с базовой Xception. Для лучших показателей Top-1, Top-3 и Top-5 Accuracy прирост составляет 0.64-3.58%, 0.25-2.36% и 0.08-1.49%, в то время как число параметров FtFNN уменьшается на 7.4-176.8 тысяч в сравнении с базовой сетью. Число выполняемых операций для обеих архитектур составляет 9.13 GFLOPS: оценки параметра *scale* из теоремы 4 отрицательны для всех наборов. Хотя при

увеличении числа классов верхняя граница становится положительной, что указывает на вычислительную эффективность FtFNN для датасетов, содержащих большее количество классов.

Таблица 6 — Значения метрики Accuracy (в %) и число параметров (млн.) для FtFNN с кодировщиком MobileNetV1.

Набор	**Top-1 макс.**	**Top-1 макс. (FtFNN)**	**Top-1 сред.**	**Top-1 сред. (FtFNN)**	**Top-3 макс.**	**Top-3 макс. (FtFNN)**	**Top-5 макс.**	**Top-5 макс. (FtFNN)**	**Парам.**	**Парам. (FtFNN)**
OFL	79.57	**86.59**	78.07 ± 1.49	**85.72± 0.67**	88.86	**91.16**	93.18	**95.70**	3.333	**3.303**
OIP	79.17	**81.27**	79.16 ± 0.02	**81.05± 0.25**	93.13	**94.22**	95.58	**96.29**	3.267	**3.251**
TFL50	92.09	**93.02**	91.79 ± 0.30	**92.79± 0.18**	98.68	**99.12**	–	–	3.234	**3.232**
TFL40	90.23	**91.37**	90.23 ± 0.13	**91.34± 0.06**	98.69	**98.86**	–	–	3.234	**3.232**
TFL30	88.9	**90.46**	88.59 ± 0.43	**90.44± 0.02**	98.09	**98.44**	–	–	3.234	**3.232**
UCM25	93.20	**94.47**	92.98 ± 0.3	**94.28 ± 0.31**	98.66	**99.30**	99.68	**99.75**	3.267	**3.241**
UCM20	91.52	**92.8**	91.35 ± 0.16	**92.74± 0.05**	97.7	**97.87**	98.0	**98.04**	3.267	**3.241**
UCM15	89.8	**92.0**	89.67 ± 0.13	**91.82± 0.09**	97.0	**98.04**	98.77	**99.0**	3.267	**3.241**
UCM10	87.0	**89.57**	86.88 ± 0.09	**89.39± 0.22**	96.19	**96.23**	97.66	**97.81**	3.267	**3.241**
INT	92.88	**94.73**	92.81 ± 0.06	**93.89± 0.60**	98.21	**98.50**	99.35	**99.52**	3.25	**3.234**

MobileNetV1

В таблице 6 для девяти обучающих наборов (см. раздел 1.4.1) приведены оценки точности классификации, полученные MobileNetV1 и построенной на ее основе FtFNN. Значения метрики Top-1 Accuracy, полученные двумя архитектурами, сравниваются на рисунке 1.5. Для лучших показателей Top-1, Top-3 и Top-5 Accuracy, полученных FtFNN, прирост составляет 0.68 -7.02%, 0.04-2.30%

и 0.07-2.52%. Число выполняемых операций для обеих архитектур составляет 1.15 GFLOPS, в то время как число параметров FtFNN уменьшается на 2.4-29.7 тысяч (аналитическая оценка параметра *scale* при этом равняется 1.0).

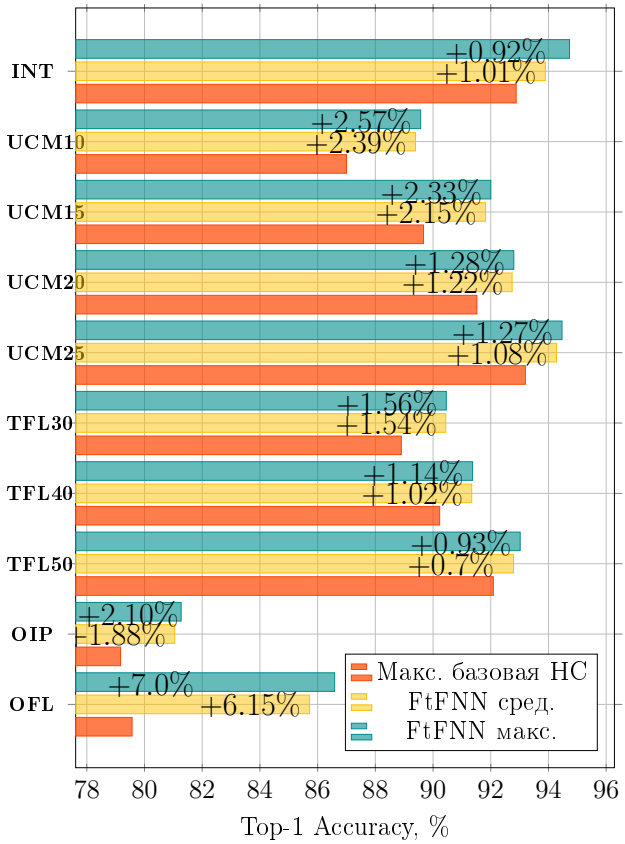

Рисунок 1.5 — Значения Top-1 Accuracy, полученные MobileNetV1 и FtFNN

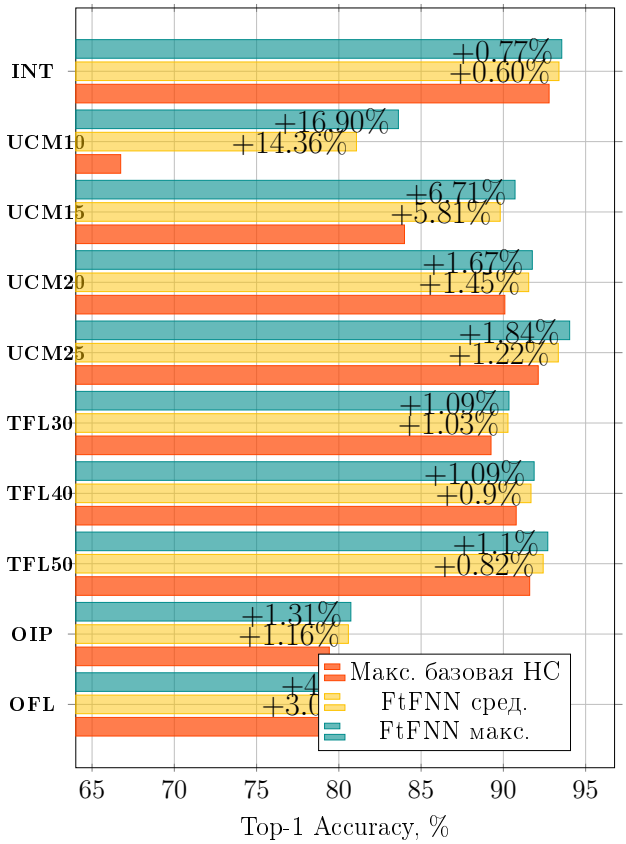

Рисунок 1.6 — Значения Top-1 Accuracy, полученные MobileNetV2 и FtFNN

MobileNetV2

В таблице 7 для девяти обучающих наборов (см. раздел 1.4.1) приведены оценки точности классификации, полученные MobileNetV2 и построенной на ее основе FtFNN. Значения метрики Top-1 Accuracy, полученные двумя архитектурами, сравниваются на рисунке 1.6. Для всех исследуемых наборов FtFNN демонстрирует более высокие показатели точности классификации в сравнении с базовой MobileNetV2. Для лучших показателей Top-1, Top-3 и Top-5 Accuracy прирост составляет 0.92-16.90%, 0.17-10.23% и 0.07-5.67%, в то время как число параметров FtFNN уменьшается на 4.5-78.6 тысяч для 6 из 9 рассмотренных

наборов (в оставшихся трех случаях их количество больше всего лишь на 8.5-9 тысяч). В случае набора OFL FtFNN вычислительно проще базовой сети как по аналитическим оценкам (экспериментальное значение $scale = 1$ меньше чем аналитическая верхняя граница 2.53), так и по экспериментальным оценкам (число выполняемых операций уменьшается с 0.613 до 0.612 GFLOPS).

Таблица 7 — Значения метрики Accuracy (в %) и число параметров (млн.) для FtFNN с кодировщиком MobileNetV2

Набор	**Top-1 макс.**	**Top-1 макс. (FtFNN)**	**Top-1 сред.**	**Top-1 сред. (FtFNN)**	**Top-3 макс.**	**Top-3 макс. (FtFNN)**	**Top-5 макс.**	**Top-5 макс. (FtFNN)**	**Парам.**	**Парам. (FtFNN)**
OFL	79.0	**83.17**	78.25 ± 0.76	**82.0± 0.87**	89.85	**92.06**	93.35	**94.58**	2.388	**2.309**
OIP	79.42	**80.73**	79.33 ± 0.09	**80.58 ± 0.21**	93.21	**93.82**	95.76	**96.59**	2.305	2.3132
TFL50	91.6	**92.7**	91.49 ± 0.15	**92.42± 0.77**	98.36	**98.91**	–	–	**2.264**	2.274
TFL40	90.78	**91.87**	90.71 ± 0.09	**91.68± 0.26**	98.36	**98.77**	–	–	2.264	**2.259**
TFL30	89.25	**90.34**	88.78 ± 0.66	**90.28± 0.08**	98.05	**98.48**	–	–	**2.264**	2.274
UCM25	92.12	**93.96**	91.96 ± 0.21	**93.34± 0.08**	98.22	**99.04**	99.36	**99.62**	2.284	**2.269**
UCM20	90.01	**91.76**	89.75 ± 0.33	**91.54± 0.21**	97.7	**97.37**	98.6	**98.63**	2.284	**2.269**
UCM15	83.99	**90.71**	80.58 ± 3.78	**89.81 ± 0.88**	90.23	**94.42**	93.57	**96.38**	2.284	**2.264**
UCM10	66.71	**83.62**	61.23 ± 4.74	**81.07 ± 2.55**	80.0	**90.23**	87.38	**93.05**	2.284	**2.264**
INT	92.78	**93.55**	92.73 ± 0.05	**93.38 ± 0.11**	98.14	**98.49**	99.25	**99.48**	2.270	**2.261**

MobileNetV3

В таблице 8 для девяти обучающих наборов (см. раздел 1.4.1) приведены оценки точности классификации, полученные MobileNetV3 и построенной

на основе ее кодировщика FtFNN. Значения метрики Top-1 Accuracy, полученные двумя архитектурами, сравниваются на рисунке 1.7. Для всех исследуемых наборов FtFNN демонстрирует более высокие показатели точности классификации в сравнении с базовой MobileNetV3. Для лучших показателей Top-1, Top-3 и Top-5 Accuracy прирост составляет 0.38-3.87%, 0.07-1.12% и 0.07-0.8%. Число параметров FtFNN ниже на 296-490 тысяч в сравнении с базовой сетью, классификатор которой состоит из последовательности сверток со 1024 каналами. FtFNN значительно вычислительно проще: число выполняемых операций уменьшается с 1.46 GFLOPS до 0.116-0.116 GFLOPS.

Таблица 8 — Значения метрики Accuracy (в %) и число параметров (млн.) для FtFNN с кодировщиком MobileNetV3

Набор	**Top-1 макс.**	**Top-1 макс. (FtFNN)**	**Top-1 сред.**	**Top-1 сред. (FtFNN)**	**Top-3 макс.**	**Top-3 макс. (FtFNN)**	**Top-5 макс.**	**Top-5 макс. (FtFNN)**	**Парам.**	**Парам. (FtFNN)**
OFL	79.42	**80.90**	78.56 ± 0.86	**80.68± 0.23**	98.75	**99.87**	93.15	**93.95**	1.634	**1.338**
OIP	78.78	**79.78**	78.25 ± 0.32	**79.17± 0.61**	93.15	**93.24**	93.22	**93.34**	1.567	**1.086**
TFL50	92.64	**93.02**	92.12 ± 0.73	**92.83 ± 0.26**	98.72	**99.23**	–	–	1.535	**0.968**
TFL40	91.23	**91.78**	90.91 ± 0.45	**91.73 ± 0.07**	98.56	**99.0**	–	–	1.535	**0.968**
TFL30	89.68	**90.61**	89.37 ± 0.43	**90.34± 0.38**	98.05	**98.17**	–	–	1.535	**0.968**
UCM25	93.65	**95.37**	93.46 ± 0.19	**95.24± 0.13**	99.42	**99.49**	99.74	**99.81**	1.551	**1.061**
UCM20	90.83	**92.95**	90.47 ± 0.35	**92.74± 0.14**	98.51	**99.58**	99.16	**99.88**	1.551	**1.061**
UCM15	86.61	**90.48**	85.91 ± 0.70	**89.78 ± 0.70**	97.70	**98.38**	99.07	**99.50**	1.551	**1.061**
UCM10	82.76	**86.42**	81.58 ± 1.07	**85.72 ± 0.60**	95.47	**96.19**	98.33	**98.90**	1.551	**1.061**
INT	94.45	**94.84**	94.39 ± 0.06	**94.68± 0.12**	98.72	**98.89**	99.46	**99.62**	1.540	**0.976**

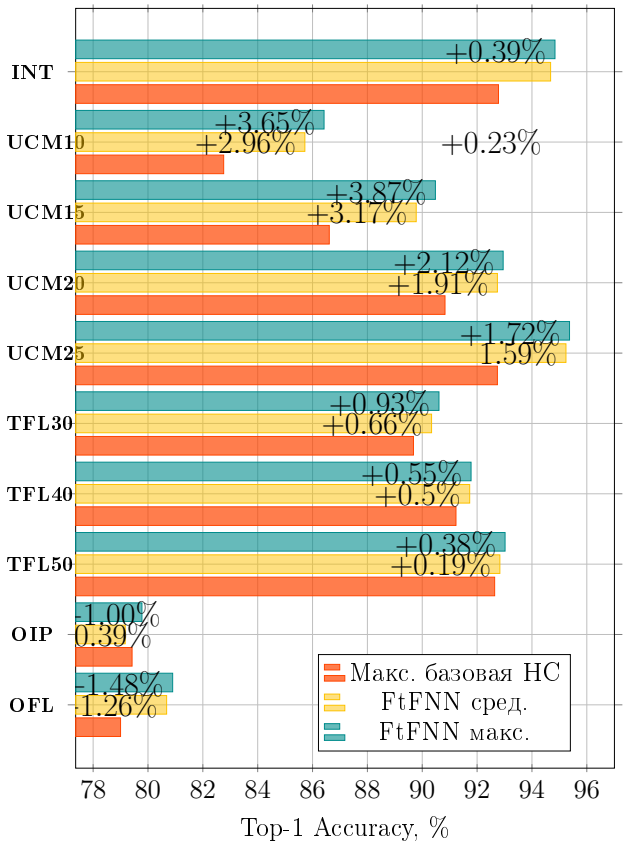

Рисунок 1.7 — Значения Top-1 Accuracy, полученные MobileNetV3 и FtFNN

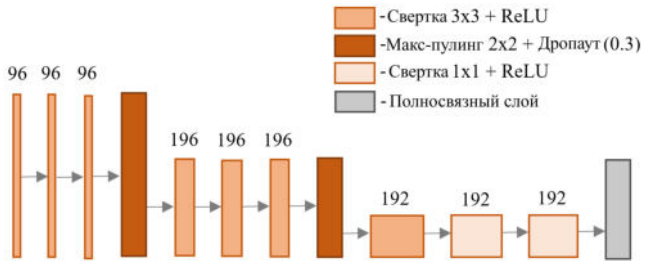

Рисунок 1.8 — Схема архитектуры FCN

1.4.5 Результаты для FtFNN с непредобученным кодировщиком

При обработке небольших наборов данных часто возникает необходимость обучать нейронную сеть с нуля, если целевые данные и данные для предобучения сильно отличаются друг от друга. Поэтому были рассмотрены конфигурации FtFNN с непредобученными кодировщиками. Исследовались сети с небольшим количеством параметров – FCN, ResNet20 (большие сети подвержены переобучению), а также наборы данных с небольшим количеством классов (для которых сети без предобучения более эффективны).

FCN

FCN – это нейросетевая архитектура, все слои которой являются сверточными (см. рисунок 1.8), а в качестве выходного классификатора используется композиция пулинга в среднем и полносвязного слоя. В таблице 9 для девяти обучающих наборов (см. раздел 1.4.1) приведены оценки точности классификации, полученные FCN и построенной на основе ее кодировщика FtFNN. Значения метрики Top-1 Accuracy, полученные двумя архитектурами, сравниваются на рисунке 1.9. Для всех исследуемых наборов FtFNN демонстрирует более высокие показатели точности классификации в сравнении с базовой FCN. Для лучших показателей Top-1, Top-3 и Top-5 Accuracy прирост составляет 1.64-2.44%, 0.25-1.01% и 0.27-0.41%. Базовая FCN вычислительно проще FtFNN: 41.12 GFLOPS против 41.13, что также согласуется с аналитическими оценками (для всех наборов верхняя граница *scale* отрицательна).

Таблица 9 — Значения метрики Accuracy (в %) и число параметров (млн.) для FtFNN с кодировщиком FCN.

Набор	**Top-1 макс.**	**Top-1 макс. (FtFNN)**	**Top-1 сред.**	**Top-1 сред. (FtFNN)**	**Top-3 макс.**	**Top-3 макс. (FtFNN)**	**Top-5 макс.**	**Top-5 макс. (FtFNN)**	**Парам.**	**Парам. (FtFNN)**
TFL50	81.69	**83.32**	80.85 ± 0.65	**82.53± 0.79**	96.34	**97.27**	–	–	1.444	**1.443**
TFL40	78.47	**80.15**	78.02 ± 0.45	**79.84± 0.32**	96.05	**96.82**	–	–	1.444	**1.443**
TFL30	75.51	**77.54**	75.41 ± 0.10	**77.40± 0.14**	95.40	**95.65**	–	–	1.444	**1.443**
CF30	80.86	**83.52**	80.81 ± 0.05	**82.66± 0.28**	95.05	**96.06**	98.58	**98.99**	**1.445**	1.463
CF25	79.58	**82.02**	79.28 ± 0.30	**81.63 ± 0.39**	94.58	**95.54**	98.45	**98.72**	**1.445**	1.463
CF20	78.19	**80.59**	77.47 ± 0.71	**79.98 ± 0.43**	94.02	**94.93**	98.22	**98.49**	**1.445**	1.463

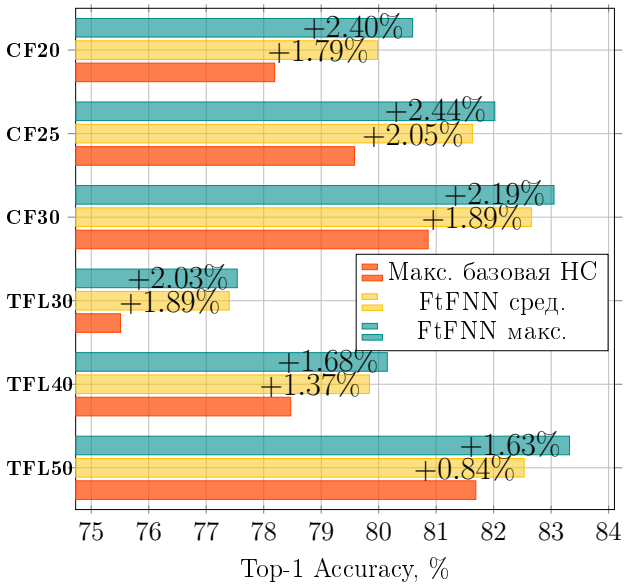

Рисунок 1.9 — Значения Top-1 Accuracy, полученные FCN и FtFNN

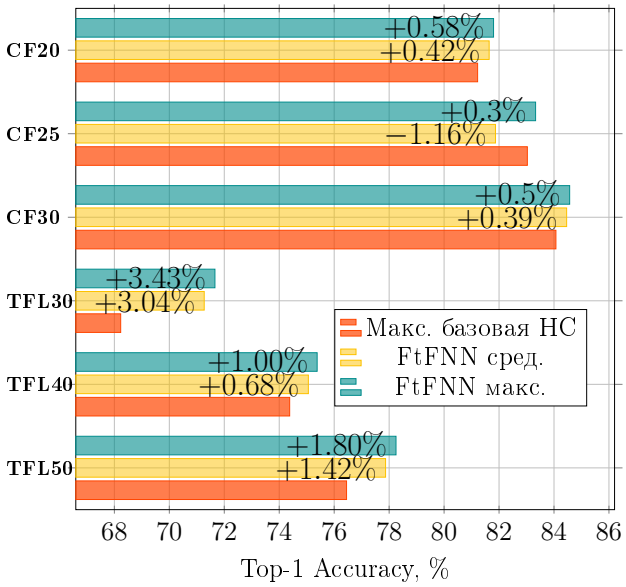

Рисунок 1.10 — Значения Top-1 Accuracy, полученные ResNet20 и FtFNN

ResNet20

В таблице 10 для девяти обучающих наборов (см. раздел 1.4.1) приведены оценки точности классификации, полученные ResNet20 и построенной на основе ее кодировщика FtFNN. Значения метрики Top-1 Accuracy, полученные двумя архитектурами, сравниваются на рисунке 1.10. Для всех исследуемых наборов FtFNN демонстрирует более высокие показатели точности классификации в сравнении с ResNet20. Для лучших показателей Top-1, Top-3 и Top-5 Accuracy прирост составляет 0.3-3.43%, 0.07-1.42% и 0.08-0.11%. Базовая ResNet20 более вычислительно проста и содержит меньше параметров, чем FtFNN: 4.03 GFLOPS против 4.04 (для всех наборов аналитическая оценка параметра *scale* отрицательна), и 274 тысяч параметров против 1.4-2.0 миллионов.

Крупные сети обычно избыточны по количеству параметров, и уменьшение их числа может повысить точность прогнозов. Однако для небольшой сети это может, наоборот, привести к значительному ухудшению результатов. Поэтому в случае ResNet20 конфигурации FtFNN, меньшие по числу параметров исходной сети, не повышают качество и точность предсказаний.

Таблица 10 — Значения метрики Accuracy (в %) и число параметров (млн.) для FtFNN с кодировщиком ResNet20 network.

Набор	**Top-1 макс.**	**Top-1 макс. (FtFNN)**	**Top-1 сред.**	**Top-1 сред. (FtFNN)**	**Top-3 макс.**	**Top-3 макс. (FtFNN)**	**Top-5 макс.**	**Top-5 макс. (FtFNN)**	**Парам.**	**Парам. (FtFNN)**
TFL50	76.45	**78.25**	76.30 ± 0.15	**77.87± 0.38**	95.70	**95.91**	–	–	**0.274**	0.425
TFL40	74.38	**75.38**	73.88 ± 0.50	**75.06± 0.31**	94.44	**95.86**	–	–	0.274	**0.273**
TFL30	68.23	**71.66**	67.28 ± 0.96	**71.27± 0.39**	93.88	**94.88**	–	–	0.274	**0.273**
CF30	84.07	**84.57**	84.03 ± 0.04	**84.46± 0.08**	95.94	**96.14**	98.95	**99.04**	**0.274**	0.276
CF25	83.03	**83.33**	82.97 ± 0.05	**81.87 ± 1.82**	95.70	**95.77**	98.80	**98.88**	**0.274**	0.278
CF20	81.22	**81.80**	81.21 ± 0.01	**81.64± 0.14**	94.83	**94.93**	98.56	**98.67**	**0.274**	0.276

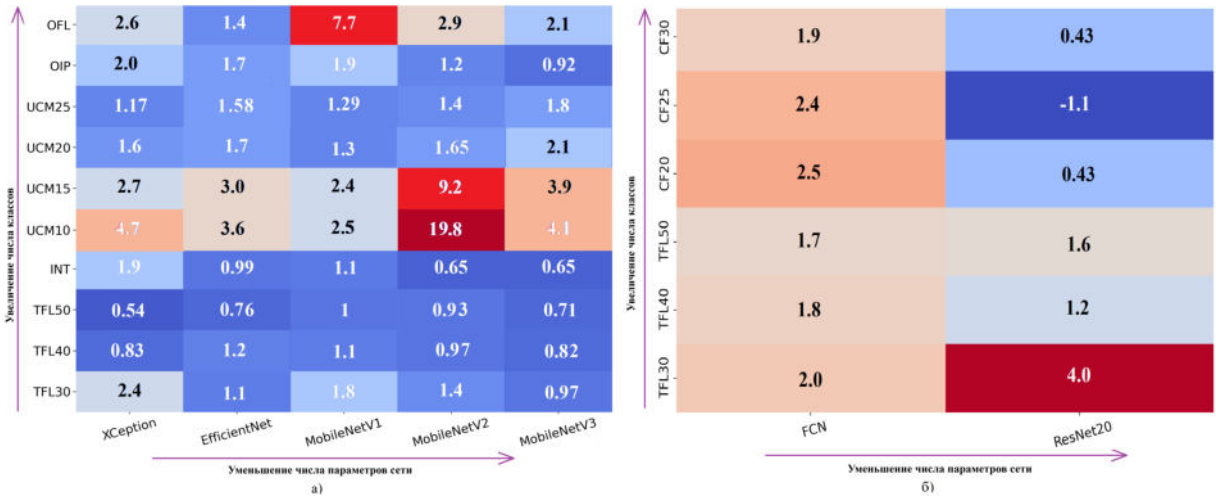

Рисунок 1.11 — Средние приросты точности по метрике Accuracy для FtFNN с предобученными (а) и непредобученными (б) кодировщиками

1.5 Оценивание точности FtFNN при разном количестве обучающих элементов в наборе

Приросты средних значений Top-1 Accuracy по сравнению с показателями базовых сетей сгруппированы в порядке убывания количества классов в набо-

Таблица 11 — Значения метрики Accuracy (в %), полученные при перекрестной проверке на данных UCM

НС	Top-1 сред.	Top-1 сред. (FtFNN)	Top-3 макс.	Top-3 макс. (FtFNN)	Top-5 макс.	Top-5 макс. (FtFNN)
			UCM 25			
EfficientNet	96.01 ± 0.45	$\mathbf{97.155 \pm 0.37}$	99.47	**99.76**	99.87	**99.9**
Xception	94.63 ± 0.38	$\mathbf{95.85 \pm 0.25}$	99.04	**99.33**	99.74	**99.82**
MobileNetV1	93.50 ± 0.33	$\mathbf{95.32 \pm 0.38}$	98.66	**99.3**	99.68	**99.75**
MobileNetV2	93.06 ± 0.57	$\mathbf{94.67 \pm 0.52}$	99.14	**99.14**	**99.76**	99.66
MobileNetV3	94.43 ± 0.70	$\mathbf{95.5 \pm 0.23}$	99.42	**99.49**	99.85	**99.86**
			UCM 20			
EfficientNet	94.40 ± 0.35	$\mathbf{95.94 \pm 0.59}$	99.28	**99.54**	99.76	**99.88**
Xception	92.29 ± 1.10	$\mathbf{93.76 \pm 0.92}$	98.62	**98.86**	99.52	**99.76**
MobileNetV1	91.72 ± 0.66	$\mathbf{93.16 \pm 0.95}$	97.8	**97.99**	99.34	**99.52**
MobileNetV2	89.56 ± 0.67	$\mathbf{91.46 \pm 0.39}$	**97.71**	97.47	98.9	**99.0**
MobileNetV3	91.48 ± 0.87	$\mathbf{92.94 \pm 0.63}$	99.09	**99.58**	99.62	**99.88**
			UCM 15			
EfficientNet	92.65 ± 0.84	$\mathbf{95.46 \pm 1.06}$	99.14	**99.61**	99.66	**99.80**
Xception	90.59 ± 1.13	$\mathbf{93.35 \pm 0.89}$	97.95	**98.66**	99.0	**99.09**
MobileNetV1	90.61 ± 2.07	$\mathbf{92.90 \pm 1.28}$	97.8	**97.99**	99.34	**99.52**
MobileNetV2	80.98 ± 4.93	$\mathbf{89.61 \pm 2.52}$	95.99	**98.8**	98.23	**99.47**
MobileNetV3	88.22 ± 1.34	$\mathbf{90.61 \pm 1.17}$	98.14	**98.38**	99.28	**99.38**
			UCM 10			
EfficientNet	87.72 ± 1.12	$\mathbf{90.44 \pm 1.16}$	97.4	**94.42**	98.71	**98.87**
Xception	85.39 ± 1.5	$\mathbf{88.4 \pm 1.65}$	96.84	**96.85**	98.57	**98.65**
MobileNetV1	84.61 ± 2.87	$\mathbf{88.39 \pm 1.51}$	96.23	**97.42**	98.28	**98.9**
MobileNetV2	63.1 ± 4.48	$\mathbf{77.53 \pm 7.06}$	86.61	**93.23**	92.52	**95.80**
MobileNetV3	80.95 ± 2.99	$\mathbf{84.40 \pm 1.33}$	96.52	**96.76**	98.14	**98.57**

ре (сверху вниз) и в порядке возрастания количества параметров кодировщика (слева направо) на рисунке 1.11. В полученных результатах можно выделить следующие закономерности: во-первых, прирост точности классификации тем выше, чем меньше элементов в обучающем наборе денных. Закономерность

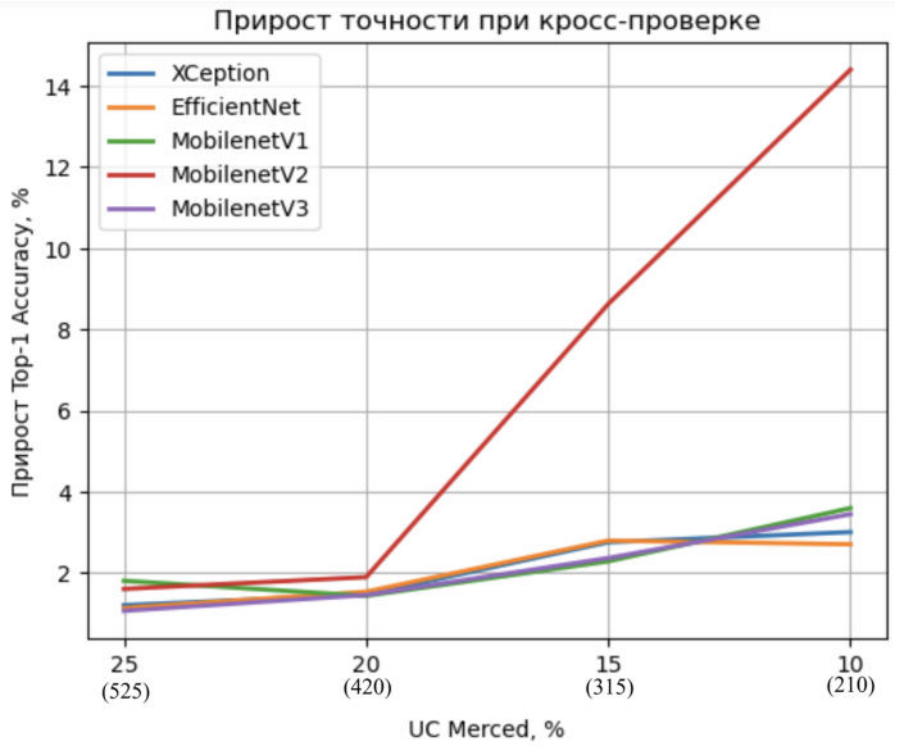

Рисунок 1.12 — Приросты средних значений Top-1 Accuracy, полученные FtFNN при перекрестной проверке на наборах UCM

можно явно проследить на наборах UCM и TFL, для которых обучающие и тестовые множества формируются из первых $n\%$ полного датасета.

Чтобы продемонстрировать общность полученных результатов, для набора UCM дополнительно была выполнена перекрестная проверка на разбиении по 4, 5, 6 и 10 фолдов данных (доля обучающих данных составляла 25%, 20%, 15% и 10% от всего набора соответственно). Результаты представлены в таблице 11.

Для всех рассмотренных кодировщиков FtFNN демонстрирует более высокие показатели точности классификации на всех наборах UCM. Прирост точности классификации по метрике Top-1 Accuracy при обработке FtFNN лежит в пределах от 1.07 до 14.4%. При этом приросты максимальных значений, Top-3 и Top-5 Accuracy равняются 0.0-6.62% и 0.03-3.28%. Результаты перекрестной проверки демонстрируют общезначимость ранее полученных результатов. FtFNN демонстрирует более высокую точность классификации при обработке небольших наборов данных и их частей по сравнению с базовыми архитектурами. Кроме того, они подтверждают зависимость прироста точности от количества элементов в обучающем наборе данных – график изменения средних значений Top-1 Accuracy представлен на рисунке 1.12.

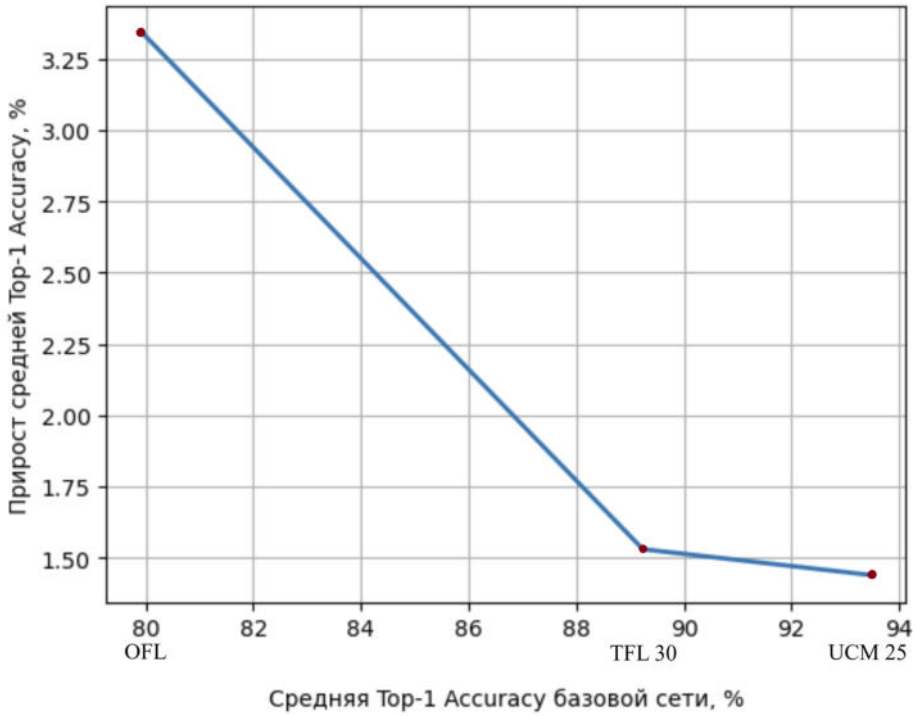

Рисунок 1.13 — Приросты средних значений Top-1 Accuracy, полученные FtFNN, для близких по числу элементов наборов (OFL, UCM 25, TFL)

Однако, помимо числа элементов, важным свойством обучающего набора, обуславливающим прирост точности классификации, является однородность. На рисунке 1.13 представлен график значений средних приростов Top-1 Accuracy в зависимости от средней точности классификации базового кодировщика для близких по числу элементов наборов OFL, UCM 25 и TFL 30. Если набор характеризуется выраженными внутриклассовыми различиями, что проявляется в более низкой точности результатов базового кодировщика, то FtFNN позволяет более существенно повысить точность обработки набора. Средний прирост Top-1 Accuracy достигает 3.34% для более неоднородного набора OFL, а для однородного UCM 25 – 1.44%.

Количество классов в наборе является фактором, способствующим повышению неоднородности данных. Поэтому по рисунку 1.11 можно заметить более высокие приросты точности классификации у наборов OFL и UCM, содержащих по 102 и 21 классов, в сравнении с остальными датасетами. Но влияние этого фактора остается косвенным – например, у наборов OIP и TFL 30, содержащих 37 и 5 классов соответственно, приросты Top-1 Accuracy близки по значениям.

1.6 Выводы

В главе представлен новый метод информирования нейросетевых классификаторов моделью факторного анализатора с аддитивным и импульсным шумами для реализации слияния глобальных многомасштабных признаков изображений. Для новой факторной вероятностной модели были доказаны идентифицируемость, а также несмещенность и состоятельность оценок параметров, полученных в ходе минимизации кросс-энтропии. Это теоретически обосновывает меньшую подверженность искажениям неросетевой архитектуры, информированной моделью такого анализатора, в особенности при недостатке обучающих данных. Также аналитически обоснован выбор стратегии реализации информирования на уровне архитектуры сети.

Новая информированная архитектура для классификации изображений FtFNN состоит из кодировщика признаков, формирующего мультимасштабные глобальные представления снимка, и информированного факторами классификатора. FtFNN демонстрирует более высокую точность классификации малых наборов изображений в сравнении со всеми рассмотренными базовыми сверточными архитектурами. Максимальные приросты Top-1, Top-3 и Top-5 Accuracy составляют 16.9%, 10.23% и 5.67%. Средний прирост Top-1 Accuracy, полученный при перекрестной проверке, достигает 14.4%.

Моделирование шума повышает точность классификации: в 45 из 72 случаев прирост точности над базовыми архитектурами был получен с использованием новой функции активации $StGeLU$, моделирующей аддитивный шум, а в 65 из 72 – при моделировании импульсного шума с помощью слоя дропаута.

Для FtFNN было доказано, что информированный классификатор может быть более вычислительно простым, чем базовые. Это подтверждается и экспериментально: количество параметров сети увеличивается по сравнению с базовым классификатором только в 10 из 72 тестов, в остальных случаях оно уменьшается – вплоть до 496 тысяч. Максимальное сокращение количества операций достигает 1.343 миллиона FLOPS.

Для FtFNN было показано, что прирост точности классификации связан с числом элементов в обучающем наборе – он тем выше, чем меньше число элементов. Важным свойством, определяющим эффективность применения FtFNN для повышения точности классификации, является однородность данных, в том

числе определяющаяся количеством классов в обучающем наборе. FtFNN демонстрирует более высокие приросты точности на более неоднородных данных (см. рисунок 1.11)

Согласно тесту Фридмана [179], при уровне значимости 0.01 разница в значениях Top-1 Accuracy, полученных базовыми архитектурами и FtFNN, является статистически значимой: в случае архитектур EfficientNet, XCeption и Mobilenets p-value не превосходит 0.001, а для FCN и ResNet p-value составляет до 0.0025. Этот результат указывает на устойчивую тенденцию к повышению точности классификации малых наборов с помощью информированной архитектуры FtFNN.

Глава 2. Информирование нейронных сетей композицией моделей конечной смеси распределений и случайного поля Маркова для сегментации неоднородных датасетов

Глава посвящена методу информирования нейронных сетей композицией вероятностных моделей для решения задачи сегментации сильно неоднородных и ограниченных по числу элементов наборов изображений. Неоднородность, проявляющаяся как высокий уровень внутриклассовых различий, часто возникает из-за влияния случайных и систематических помех, обусловленных как свойствами оборудования, например, разрешением съемки, так и внешними факторами, такими как погодные условия или освещенность сцены.

Ограниченность набора по числу элементов или его характеристик, как правило, усиливает неоднородность. Поэтому задача сегментации является более сложной, в сравнении с классификацией, поскольку на каждый классифицируемый элемент (пиксель) приходится существенно меньше данных (его собственная яркость и яркости пикселей-соседей), чем при классификации снимка целиком. Ярко это проявляется в рамках обучения по нескольким примерам (англ. few-shot learning) – работ, посвященных решению задачи сегментации существенно меньше, чем классификации [180—182].

Для сегментации неоднородных наборов требуется больше дополнительных данных, чем при решении задачи классификации. Решением может стать объединение дополнительной информации из нескольких вероятностных моделей, описывающих локальные (на уровне пикселя) признаки изображения, их яркостные и пространственные свойства. Яркостные свойства пикселей могут быть описаны [183] конечной смесью вероятностных распределений. Пространственные же взаимосвязи между ними могут быть промоделированы с помощью случайного поля Маркова [184]. В главе исследуются аналитические свойства выбранных моделей при обработке неоднородных данных. Также аналитически определяется и обосновывается выбор способа информирования каждой из выбранных моделей нейросетевого сегментатора.

2.1 Постановка задачи

Пусть дан неоднородный набор изображений $\mathbb{X} = \{X_{ij}^l\}_{l=\overline{1,N}}$, $i = \overline{1, H_S}, j = \overline{1, W_S}$, где H_S и W_S – высота и ширина в пикселях обрабатываемого изображения. Обозначим $\Theta(X^l)$ признаки пикселей изображения X^l, получаемые из вероятностной модели, описывающей их яркостные свойства, $\Theta : \mathbb{R}^{\theta} \to \mathbb{R}^K$, а $Q(X^l)$ – получаемые моделированием пространственных взаимосвязей между пикселями, $Q(X^l)$, $Q : \mathbb{R}^q \to \mathbb{R}^K$.

Требуется разработать НС модель $F = Q \circ f \circ \Theta$, где $f(\cdot)$ – базовый сегментатор, а символ «$\circ$» означает композицию отображений, повышающую информативность набора $\mathbb{X}$ за счет информирования НС $f(\cdot)$ признаками $\Theta(X^l)$ и $Q(X^l)$ для повышения вероятности правильной сегментации X^l, то есть присвоения каждому его пикселю X_{ij}^l метки $Y_{ij}^l = \arg\max\limits_{k=1,\dots,K} \mathbb{P}(X_{ij}^l \mid k)$:

$$\mathbb{P}(Q \circ f \circ \Theta(X_{ij}^l) = Y_{ij}^l) \geqslant \mathbb{P}(f(X^l) = Y_{ij}^l).$$

Схема НС модели, рассматриваемой в решаемой задаче, представлена на рисунке 2.1.

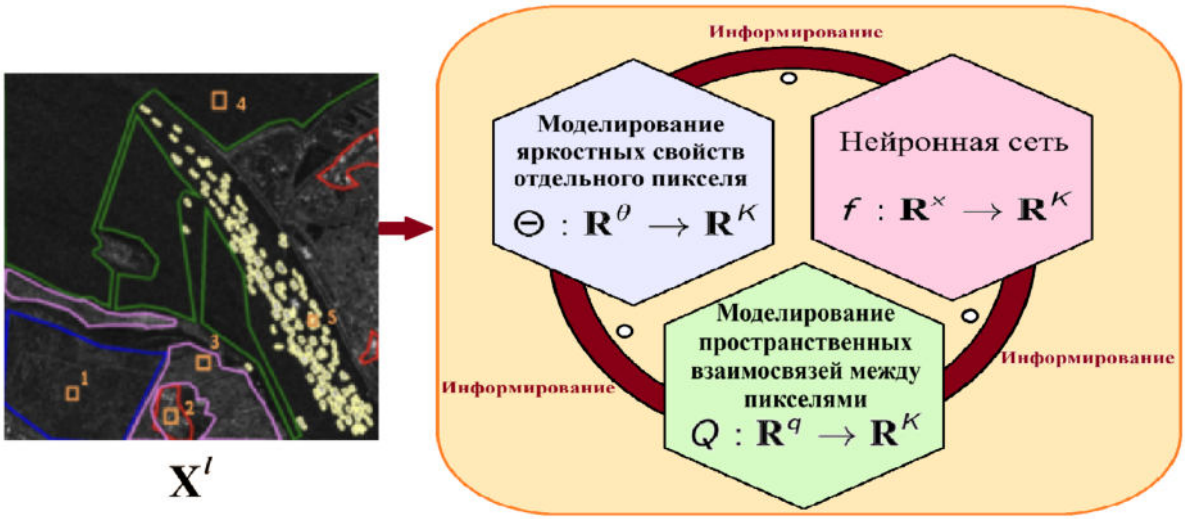

Рисунок 2.1 — Концепт НС модели, информированной композицией вероятностных моделей для сегментации неоднородных наборов изображений

2.2 Информирование моделью конечной смеси вероятностных распределений

Яркость пикселя изображения считается случайной величиной, поскольку ее значение определяется отражающими свойствами поверхности и случайным уровнем спекл-шума (шума зернистой структуры, вызванного интерференцией когерентных волн, отраженных от поверхности). Такое представление верно как для обычных изображений, так и для специализированных, например, спутниковых снимков [185]. Таким образом, изображение представляет совокупность выборок из K различных случайных величин, определенных на вероятностном пространстве $(\Omega, \mathcal{F}, \mathbb{P})$, где $\Omega = \{0,1,\ldots,N_{br}\}$ (N_{br} – максимальное значение яркости пикселя обрабатываемого изображения, которое при нейросетевой обработке, как правило, равняется 255), а $\mathcal{F}$ и $\mathbb{P}$ – соответствующим образом заданные сигма-алгебра событий и вероятностная мера. Количество случайных величин K соответствует числу представленных на снимке типов поверхностей. Такая структура данных называется конечной смесью вероятностных распределений. Для упрощения вычислений нередко используют аппроксимацию нормальными смесями, распределение компонент которых описывается параметрами сдвига и масштаба (a_k, σ_k), причем для всех k величины $a_k \in \mathbb{R}$, $\sigma_k > 0$ [137]. Плотность распределения K-компонентной смеси нормальных законов описывается формулой:

$$h(x) = \sum_{k=1}^{K} p_k \, \varphi\left(\frac{x - a_k}{\sigma_k}\right), \tag{2.1}$$

где $\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ – плотность распределения стандартного нормального закона, а $p_k \in [0,1]$ – весовой коэффициент k-й компоненты смеси, и $\sum_{k=1}^{K} p_k = 1$.

Моделирование изображения конечной нормальной смесью означает вычисление для каждого пикселя $\hat{X}_l$, $l = i \cdot W_X + j$, вероятности принадлежать каждой компоненте смеси. Вычисление этих вероятностей требует знания параметров компонент смеси. Классическим методом их оценки является итерационный алгоритм вычисления оценок максимального правдоподобия – Expectation-Maximization (EM) [186], и его разнообразные модификации [187; 188].

Изображение может обрабатываться EM алгоритмом как целиком, так и по отдельным фрагментам, что более вычислительно эффективно и позволяет сохранять мелкие детали [122]. Наиболее предпочтительным по производительности для реализации этапа предобработки данных способом согласования распределений компонент является их фиксирование во всех фрагментах – изменяются только их весовые коэффициенты p_k (см. формулу (2.1)). Для этого на основе эмпирического анализа изображения для каждой компоненты смеси должны выбираются эталонные области для оценки по ним значений a_k и σ_k [189]. Пары точек (a_k, σ_k) образуют сетку в пространстве параметров.

В случае, когда неизвестными являются только весовые коэффициенты компонент смеси, для их оценки используют сеточный EM алгоритм [190; 191]. В этом алгоритме параметры вычисляются в ходе максимизации функции вида:

$$\ln L(\mathbb{X}, p_k, a_k, \sigma_k) = \sum_{k=1}^{K}\sum_{i=1}^{n} g_{ik} \ln p_k + \\ + \sum_{k=1}^{K}\sum_{i=1}^{n} g_{ik} \left(\ln \varphi \left(\frac{\hat{X}_i - a_k}{\sigma_k} \right) - \ln g_{ik} \right), \tag{2.2}$$

где $\mathbb{X} = (\hat{X}_1 \dots \hat{X}_n)$ – значения выборки, Θ – вектор известных значений параметров компонент смеси, а g_{ik} – апостериорная вероятность соответствия элемента выборки $\hat{X}_l$ k-й компоненте смеси. На r-й итерации алгоритма значения $p_k^{(r)}$ могут быть вычислены по формулам, полученным методом неопределенных множителей Лагранжа:

$$p_k^{(r)} = n^{-1} \sum_{i=1}^{n} g_{ik}^{(r)}, \quad g_{ik}^{(r)} = \frac{p_k^{(r-1)} \varphi \left(\frac{\hat{X}_l - a_k}{\sigma_k} \right)}{\sum_{l=1}^{K} p_l^{(r-1)} \varphi \left(\frac{\hat{X}_l - a_l}{\sigma_l} \right)}, \tag{2.3}$$

Согласно [190] оценки, полученные сеточным EM алгоритмом, сходятся к глобальному максимуму функции (2.2). Тогда для каждого l-го пикселя изображения вероятности принадлежать каждой из компонент смеси $p^*(X_l)$ имеют вид:

$$\left(p_1 \, \varphi \left(\frac{\hat{X}_l - a_1}{\sigma_1} \right), \dots, p_K \, \varphi \left(\frac{\hat{X}_l - a_K}{\sigma_K} \right) \right). \tag{2.4}$$

Аналитически можно показать, что если использовать признаки (2.4) в качестве дополнительных входных данных сети, то такое информирование вероятностями $p^*(\hat{X}_l)$ способно повысить точность предсказаний нейронной сети

на сильно неоднородном наборе данных. Введем понятие шага h интерполяционной сетки, составленной из элементов обучающего набора. Под этим значением будем понимать среднее арифметическое расстояний между элементами обучающего набора, рассчитанное, например, с использованием евклидовой метрики. Если в наборе мало элементов или если они сильно отличаются друг от друга, среднее расстояние будет выше, чем для большого и однородного набора данных.Если данные нормализованы, то при $h \geqslant 1$ набор можно считать неоднородным, так как элементы разделены расстоянием, превышающим их среднеквадратичное отклонение. Этот случай соответствует умеренно большим h, когда этот параметр не устремлен к бесконечности.

Теорема 5. *Пусть $f(x)$ – полносвязная однослойная нейронная сеть, использующая для восстановления дважды дифференцируемой функции $F(x)$, а h – среднее расстояние меду нормализованными элементами обучающего набора. Тогда обогащение входных данных с помощью конечной смеси нормальных распределений, а именно представление их в виде вектора $p^*(x)$, уменьшает погрешность восстановления $F(x)$: $\mathbb{E}(\int(F(x) - f(p^*(x)))^2 dx) \leqslant M \cdot h^4$, где M – константа. Эта погрешность в случае неоднородных наборов данных, соответствующем $h \geqslant 1$, имеет меньший порядок в сравнении с ошибкой неинформированной полносвязной сети.*

Доказательство. Восстановление функции $F(x)$ можно рассматривать как задачу интерполяции. Однослойная $f(x)$ реализует линейную интерполяцию $F(x)$. Ее ошибка интерполирования $|F(x) - f(x)|$ на отрезке $[0,h]$ не превосходит выражения $M_1 \cdot h^2$, где M_1 – константа. Если перед обработкой НС входные данные обогащаются с помощью конечной смеси нормальных законов, то на вход $f(\cdot)$ подается не элемент x, а выражение (2.4). Сравним ошибки смесевой и линейной интерполяций при не слишком больших h, таких что $h \geqslant 1$, но не стремится к бесконечности. Известно [192], что для смеси интегральная средняя квадратичная ошибка $\mathbb{E}(\int(F(x) - f(x))^2 dx)$ при восстановлении дважды дифференцируемой $F(x)$ на отрезке $[0,h]$ задается полиномом, максимальная степень которого не превосходит 4: $\mathbb{E}(\int(F(x) - f(x))^2 dx) \leqslant M \cdot h^4$, где M – константа. Интегральная средняя квадратичная ошибка линейной интерполяции может быть получена из $|F(x) - f(x)|$ возведением в квадрат и интегрированием по $[0,h]$ – тогда она не превосходит $M_2 \cdot h^5$, где M_2 – константа. Если обучающий набор неоднороден, это означает, что $h \geqslant 1$. В таком случае, порядок h^5 выше,

чем порядок h^4. Это означает, что в случае неоднородных наборов обогащение входных данных с помощью смесей может повысить точность восстановления целевой функции. □

Теорема 5 рассматривает поведение ошибок интерполяции при умеренно больших интервалах разбиения $h \geqslant 1$, а не их асимптотическое поведение при $h \to 0$ или $h \to \inf$, по причине того что этот случай соответствует практической задаче обработки неоднородного набора данных. Теорема 5 показывает, что использование вероятностей компонентов смеси для восстановления целевого распределения с использованием нейронной сети способно привести к уменьшению ошибки интерполяции по сравнению с обычной полносвязной сетью. Это обосновывает применение модели смеси для информирования НС в этой задаче. Кроме того, теорема 5 устанавливает, что информирование НС моделью конечной смеси для повышения точности обработки неоднородных данных должно быть реализовано на уровне входных признаков.

2.3 Информирование с помощью случайного поля Маркова в форме квадродерева

Пусть задан граф $\zeta(\mathcal{E}, V)$, где $\mathcal{E}$ – множество ребер, V – множество вершин. Пусть также определено вероятностное пространство $(\Omega, \mathcal{F}, \mathbb{P})$, где Ω – конечное множество всех возможных конфигураций ω, а конфигурацией называется некоторое присвоение всем вершинам графа ζ меток y из конечного множества Y, а $\mathcal{F}$ и $\mathbb{P}$ – сигма-алгебра событий и вероятностная мера. Соответственно, случайным полем Маркова χ, определенном на ζ, называется вероятностная модель, удовлетворяющая следующим свойствам [138]:

– для любого $\omega \in \Omega$: $\mathbb{P}(\chi = \omega) > 0$;
– для каждого v,$r \in \mathcal{E}$ и $\omega \in \Omega$: $\mathbb{P}(\chi_v = \omega_v | \chi_r = \omega_r, r \neq v) = \mathbb{P}(\chi_v = \omega_v | \chi_r = \omega_r, r \in \zeta_v)$, где ζ_v множество связанных ребрами с v узлов графа ζ.

Двумерная пиксельная решетка является традиционной графовой структурой, используемой при описании изображения. В ней ребрами соединены не

более восьми соседних элементов – по горизонталям, вертикалям и двум диагоналям. По такой графовой структуре может быть построено случайное поле Маркова. Однако она не учитывает иерархические связи между пикселями в разных пространственных разрешениях, которые особенно важны для снижения влияния шума на результат обработки снимка. Среди графовых моделей, учитывающих многомасштабные взаимосвязи, особое место занимает квадродерево.

Квадродерево – это графовая структура пространственных данных из h расположенных друг над другом слоев. Квадродерево можно представить в виде пирамиды изображений разного пространственного разрешения $S_0 \dots S_{h-1}$, где S_0 – нижний слой дерева («листья»). Эта структура реализует концепцию пространственного бинарного поиска: каждый узел $s \in S_l$ соответствует определенному пикселю в пирамиде изображений и связан с одним родительским узлом s^- в предыдущем (верхнем) слое и с четырьмя дочерними узлами s^+ в следующем слое (за исключением слоя листьев). Взаимосвязи между узлами, заданные указанным способом, обладают марковским свойством: связаны только ближайшие соседи, в том числе в соседних слоях, но нет связей «через слои». Квадродерево называется пространственно-иерархическим, если для каждого узла дополнительно определено множество $\{s^* \in S_l, s \in S_l, s^{\preceq} s\}$, состоящее из соседних узлов s в слое S_l [193].

Таким образом, квадродерево представляет собой граф $Q_G = \{\mathcal{E}, V\}$, состоящий из $|V| = \sum_{i=0}^{h-1} \frac{H_X}{2^i} \cdot \frac{W_X}{2^i}$ узлов и $|\mathcal{E}| = \sum_{i=0}^{h-1} (\frac{H_X}{2^i} - 2) \cdot (\frac{W_X}{2^i} - 2) \cdot 9 + 12(\frac{H_X}{2^i} + \frac{W_X}{2^i} - 4) + 4 \cdot 4$ ребер. Внутри одного слоя узел связан ребрами не более чем с восьмью соседними элементами – с пятью, если узел расположен на границах изображения, и с тремя – если в углах (см. рисунок 2.2). Одновременно каждый узел из слоя S_i также связывается с одним узлом из слоя S_{i+1} и не более чем с четырьмя узлами из слоя S_{i-1} (см. рисунок 2.3). Если это множество содержит только один элемент s^*, то узлы в слое S_l расположены в виде цепи Маркова.

Пример построения квадродерева с числом слоев $h = 3$ приведен на рисунке 2.4. Исходное изображение размера $N_h \times N_h$ помещается в слой S_0, а слои $\overline{1,h-1}$ формируются из S_0 с помощью усредняющего пулинга с ядром $2^{h-i} \times 2^{h-i}$, $i = \overline{1,h-1}$. Вместо значений яркости исходного изображения в узлы квадродерева могут быть помещены значения вероятностей классов пикселей. Эти вероятности могут быть уточнены с учетом системы связей в пирами-

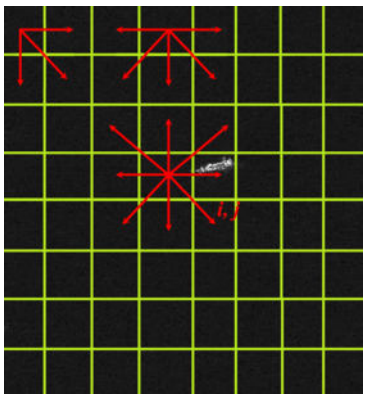

Рисунок 2.2 — Система связей узлов, расположенных в одном слое квадродерева (граф-решетка)

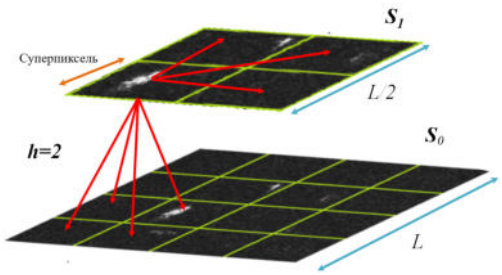

Рисунок 2.3 — Система связей узлов, расположенных в разных слоях квадродерева, $h = 2$

дальной структуре квадродерева с помощью специальной процедуры на основе байесовского вывода [193].

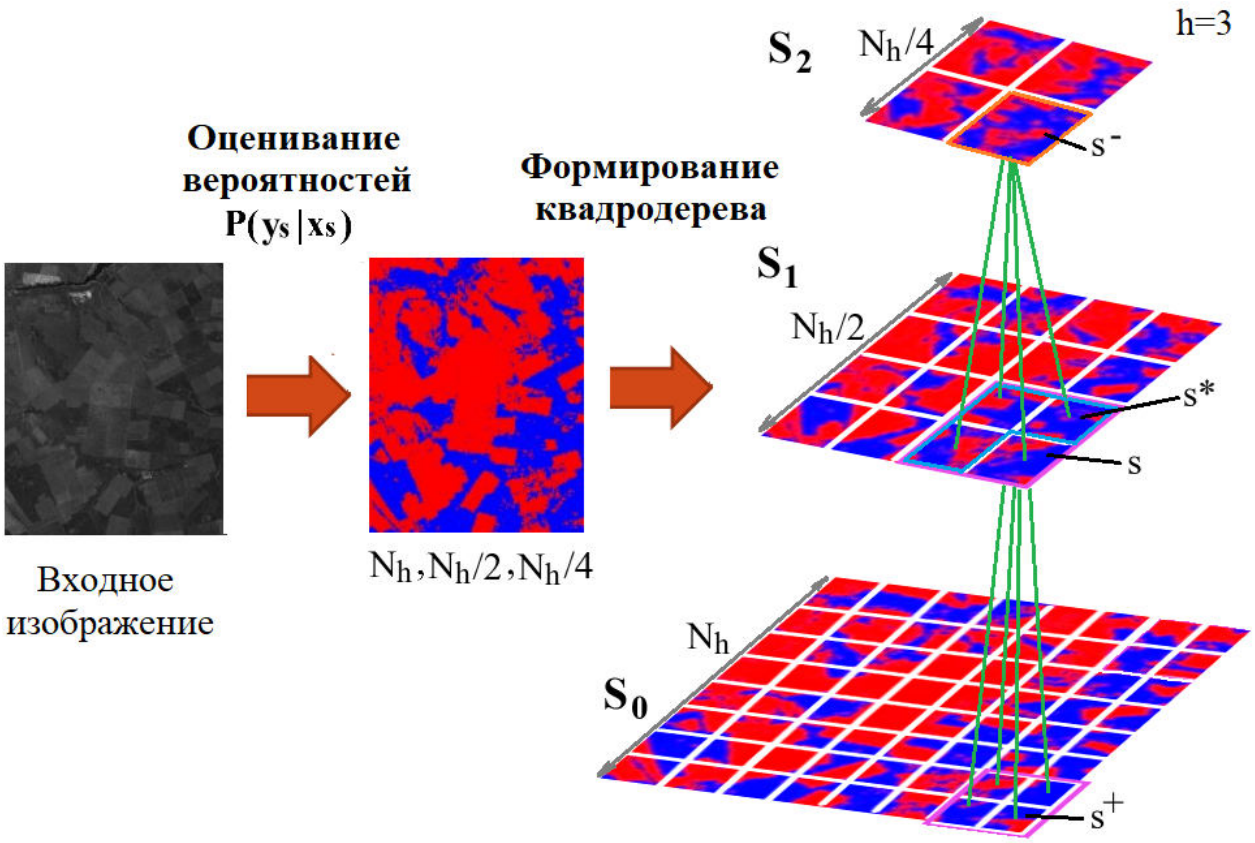

Рисунок 2.4 — Пример формирования квадродерева высоты $h = 3$

Пусть y_s – новая метка класса пикселя s, а X_s – значение яркости. Для каждого пикселя s вероятности $P(y_s \mid \mathbb{X})$, $\mathbb{X} = \{X_s\}_{s \in S}$ формируются из вероятностей $P(X_s \mid y_s)$, оцениваемых, например, с помощью нейросети. Векторы контекста $C_s = \{s^-\} \cup \{r \in S_l, r \preceq s\}, s \in \{S_0 \dots S_{h-1}\}$ и признаков всех

потомков пикселя s, $D_s = X_s, s \in S_0$ используются для описания взаимоотношений между s, s^-, s^+, s^*.

Вначале для каждого слоя вероятности содержать пиксели класса y_s определяются формулой:

$$P_{y_s} = \sum_{y_{s^-} \in \Omega} P_{y_s} \cdot P(y_s \mid y_{s^-}). \tag{2.5}$$

Затем на их основе вычисляются условные вероятности следующего вида:

– $P(y_s \mid D_s)$ – вероятность пикселю s соответствовать классу y_s в зависимости от классовой принадлежности его дочерних узлов:

$$\begin{cases} \prod\limits_{t \in s^+} \sum\limits_{y_t \in \Omega} \frac{P(y_t|D_s) \cdot P(y_t|X_s)}{P(y_t)}, s \in S \setminus S_0, \\ P(y_s \mid X_s), s \in S_0, \end{cases} \tag{2.6}$$

где величина $P(y_s \mid X_s)$ пропорциональна $P(X_s \mid y_s) \cdot P(y_s)$.

– $P(y_s \mid y_{s^-}, y_{s^*}, D_s)$ – вероятность пикселю s соответствовать классу y_s в зависимости от классовой принадлежности его дочерних узлов, родительского узла и узлов-соседей ($S_{H-1} = S \setminus S_{h-1}$):

$$\begin{cases} \frac{P(y_s|D_s) \cdot P(y_s|y_{s^*}) \cdot P(y_{s^*})}{P(y_s)^2}, s \in S_{h-1}, \\ \frac{P(y_s|D_s) \cdot \prod_{r \in \{s^-, s*\}} P(y_s|r) \cdot P(r)}{P(y_s)^2}, s \in S_{H-1}, \end{cases} \tag{2.7}$$

Наконец, используя представленные выше формулы (2.5)–(2.7), вероятности $P(y_s \mid \mathbb{X})$ могут быть вычислены с помощью следующего выражения:

$$\begin{cases} \sum\limits_{y_{s^-}, y_{s^*} \in \Omega} P(y_s \mid y_{s^-}, y_{s^*}, D_s) \cdot P(y_{s^*} \mid \mathbb{X}), s \in S_0, \\ \sum\limits_{y_{s^-}, y_{s^*} \in \Omega} P(y_s \mid y_{s^-}, y_{s^*}, D_s) \cdot P(y_{s^*} \mid \mathbb{X}) \cdot P(y_{s^-} \mid X). \end{cases} \tag{2.8}$$

Максимальный элемент вектора вероятности $P(y_s \mid \mathbb{X})$ определяет метку класса y_s. Обработка с помощью квадродерева уменьшает уровень шума за счет использования иерархических связей между изображениями в его слоях для удаления аномальных зашумленных пикселей и поддержания локальной однородности изображения.

Рассмотрим свойства поля Маркова в виде квадродерева. Во-первых, можно показать, что обработка изображений с использованием квадродерева, описываемая формулами (2.5)–(2.8), близка к методам нейросетевой обработки.

Теорема 6. *Вычисление вероятностей классов вершин квадродерева с помощью байесовского алгоритма, определяемого формулами (2.5)–(2.8), для элементов изображения X эквивалентно применению к ним предобученной графово-сверточной нейронной сети с фиксированными весами.*

Доказательство. Для доказательства утверждения необходимо показать, что хотя бы одна из формул (2.5)–(2.8) включает в себя операцию линейной графовой свертки на векторе всех узлов квадродерева x:

$$y = A \cdot x \cdot B, \tag{2.9}$$

где A – матрица смежности квадродерева или его подграфа, а B – матрица весов. В таком случае эти формулы можно считать реализацией графовой сети.

Рассмотрим формулуе (2.7), потому что ее результат напрямую используется при формировании выходной метки класса для узла дерева в формуле (2.8). Пусть в формуле (2.7) значения меток y_s,y_{s^-},y_{s^*} фиксированы. Тогда $P(y_s \mid y_{s^-})$ и $P(y_s \mid y_{s^*})$ – это двумерные нормированные матрицы, выражающие пространственные зависимости между элементами квадродерева s,s^-, s^*. Элементы этих матриц равны нулю за исключением значений, стоящих на пересечении i и j строки и столбца, где i –индекс s в векторе g, а j – s^- или s^*. Тогда $P(y_s \mid y_{s^-})$ и $P(y_s \mid y_{s^*})$ – это матрицы смежности, описывающие связи с предшественниками как внутри одного слоя квадродерева, так и в предшествующих слоях. Аналогичные выводы можно сделать для всех комбинаций значений меток y_s, y_{s^-}, y_{s^*}, и значит формула (2.7) содержит операцию графовой сверки. При этом значения $P(y_s \mid y_{s^-}), P(y_s^-), P(y_s)^{-2}, P(y_s \mid y_{s^*}), P(y_{s^*})$ задаются из априорных соображений и не изменяются в процессе обхода дерева. Значит, квадродерево можно представить как графово-сверточную сеть с предобученными фиксированными весами.

□

Согласно теореме 6 информирование квадродеревом следует реализовать на уровне архитектуры сети.

Композиция нейронных сетей и квадродеревьев ранее применялась для обработки оптических изображений [194] и радиолокационных снимков, полученных COSMO-SkyMed [195]. Однако в этих статьях наборы были достаточно большими для обучения U-Net [196] и других крупных нейросетевых архитектур, а случай ограниченных наборов данных не рассматривался и информирование на уровне входных признаков не требовалось.

В модели квадродерева для вычисления вероятностей классов элементов поля выполняется как прямой проход по слоям дерева – от S_0 к S_{h-1}, так и обратный, граф Q_G является неориентированным.

Теорема 7. *Поле Маркова, построенное по графу Q_G, обладает свойством эргодичности.*

Доказательство. Согласно определению модели граф Q_G является неориентированным. Это означает, что если в Марковском поле, построенном по графу Q_G, вероятность перехода из узла i в узел j $p_{i,j} \neq 0$, то также возможно совершить переход из j в i и $p_{j,i} \neq 0$ (при этом также будем считать, что $p_{i,i} \neq 0$). Таким образом, поле Маркова, построенное на Q_G не содержит поглощающих состояний.

Рассмотрим теперь случай, когда узлы i и j не связаны ребрами напрямую. Предположим, что i и j принадлежат одному и тому же слою дерева S_k. Поскольку количество узлов $|V|$ графа Q_G конечно, то конечно и число элементов в S_k. Тогда $\exists t$, такое что состояние j из состояния i всегда можно достичь за t шагов, перемещаясь по графу только в горизонтальном и вертикальном направлениях.

Предположим теперь, что i и j принадлежат различным слоям дерева: $i \in S_k$, $j \in S_l$ и пусть для определенности $k < l$ (для случая $k > l$ доказательство аналогично). Обозначим $i^* \in S_k$ узел, которого можно достичь из узла j переместившись из S_l в S_k по иерархическим связям между слоями за $k - l$ шагов. Согласно доказательству для узлов, принадлежащих одному слою дерева, $\exists t$, такое что i^* можно достичь из i с ненулевой вероятностью за t шагов. Таким образом, из состояния i можно с ненулевой вероятностью достичь состояния j за конечное $k - l + t$ число шагов.

Тогда по достаточному условию эргодичности [197] Марковское поле, построенное по графу Q_G, обладает свойством эргодичности. □

Свойство эргодичности делает перспективным применение модели квадродерева для описания глобальных признаков снимков в задачах обнаружения разномасштабных элементов. Оно означает существование стационарного распределения, которое позволяет перенести закономерности, выделенные для крупных объектов в слоях более низкого разрешения $S_1, \ldots, S_{h-1}$ на малые

объекты в слоях более высокого разрешения и наоборот, что способно повысить точность их выделения, особенно когда данные неоднородны или сильно не сбалансированы.

2.4 Архитектура PrINN

Согласно теореме 5, информирование признаками конечной смеси вероятностных распределений должно быть реализовано на уровне входных признаков в задаче обработки неоднородного набора. Информирование же моделью квадродерева, согласно теореме 6, должно быть реализовано отдельным архитектурным блоком.

Предложена архитектура для сегментации изображений Probability Informed Neural Network (PrINN) (см. рисунок 2.5), которая реализует эту концепцию информированная композицией двух вероятностных моделей, описанных в разделах 2.2 и 2.3. При информировании смесью на уровне признаков (левый блок на рисунке 2.5) на вход нейронной сети помимо яркостей пикселей также передаются их вероятности соответствовать каждой из K компонент смеси нормальных законов. Информирование квадродеревом (правый блок на рисунке 2.5) реализовано как пост-обработка (ансамблирование) сегментированных нейронной сетью $f(\cdot)$ снимков.

В PrINN сегментация изображений нейронной сетью проводится по перекрывающимся фрагментам (патчам) изображения, то есть, задача сегментации сводится к задаче классификации фрагментов. Традиционно для сегментации используют нейросетевые архитектуры, обрабатывающие изображение попиксельно – примером является нейронная сеть U-Net. Однако при работе с небольшими наборами данных обработка по фрагментам имеет ряд преимуществ. Во-первых, обучающий набор данных для сегментации по фрагментам может быть меньше, чем для попиксельной, поскольку в нем должны быть только однородные фрагменты, не содержащие сильно вариативных границ классов. Во-вторых, использование прочных пространственных взаимосвязей при обработке фрагментов снижает влияние шума на результат сегментации. Наконец, если изображения в целевом наборе данных не размечены, что характерно для реальных сценариев, для обучения по фрагментам нанести метки классов ока-

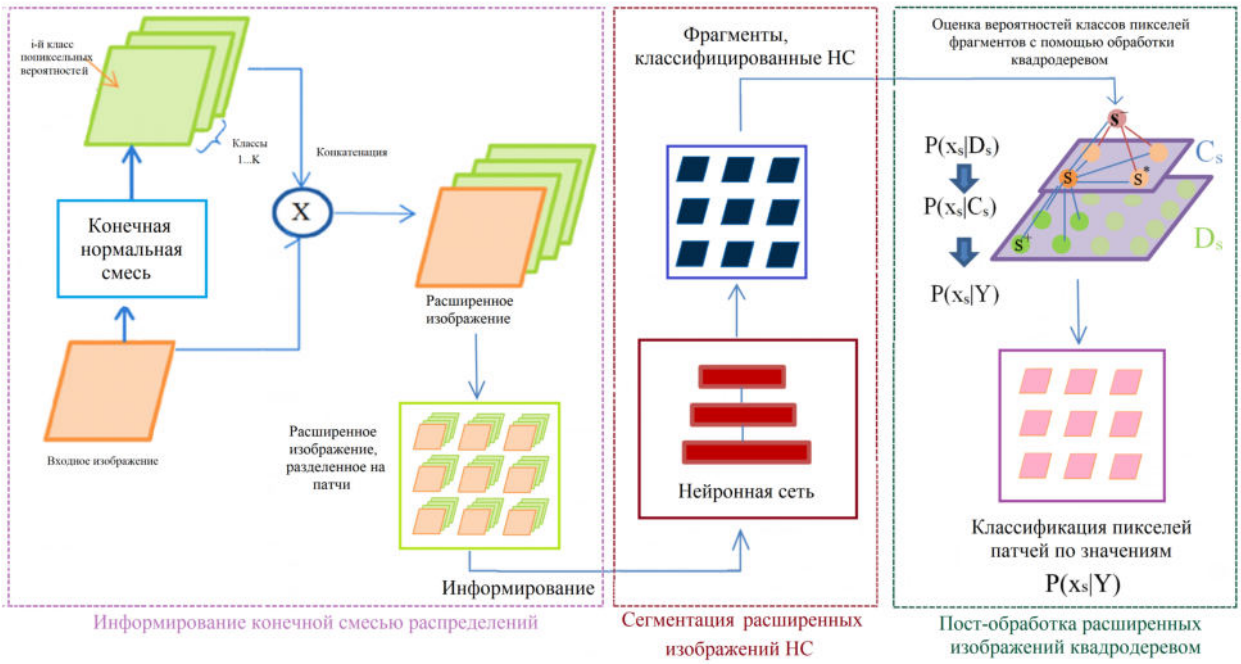

Рисунок 2.5 — Архитектура PrINN

зывается существенно быстрее и проще, чем для попиксельной сегментации. В первом случае нужно лишь вырезать несколько прямоугольных элементов из однородных областей изображения и разделить их на фрагменты, а во втором — выполнить трудоемкую и длительную задачу по выделению контуров однородных областей.

Было рассмотрено несколько архитектур для сегментации изображения по фрагментам:

– сверточная сеть (см. рисунок 2.6 (а)) с «полным» поканальным вниманием на основе скалярного произведения (см. рисунок 2.6 (г)) [198] и одной обратной связью;
– сверточная сеть с «усеченным» поканальным вниманием (см. рисунок 2.6 (д)) и одной обратной связью;
– сверточная сеть (см. рисунок 2.6 (б)) с «полным» поканальным вниманием (см. рисунок 2.6 (г)) и несколькими обратными связями;
– Vision Transformer (ViT) [28] (см. рисунок 2.6 (в)).

Схемы нейросетевых архитектур и блоков внимания представлены на рисунке 2.6 (значения гиперпараметров приведены в разделе 2.5). На рисунке 2.6 (а) представлена базовая свёрточная сеть с механизмом внимания («Малая сеть»), состоящая из двух свёрточных блоков, двух полносвязных слоёв размером $dSize$ и блока внимания. Каждый свёрточный блок содержит два свёрточных слоя с одинаковым количеством каналов, равным $Cnv1$ или

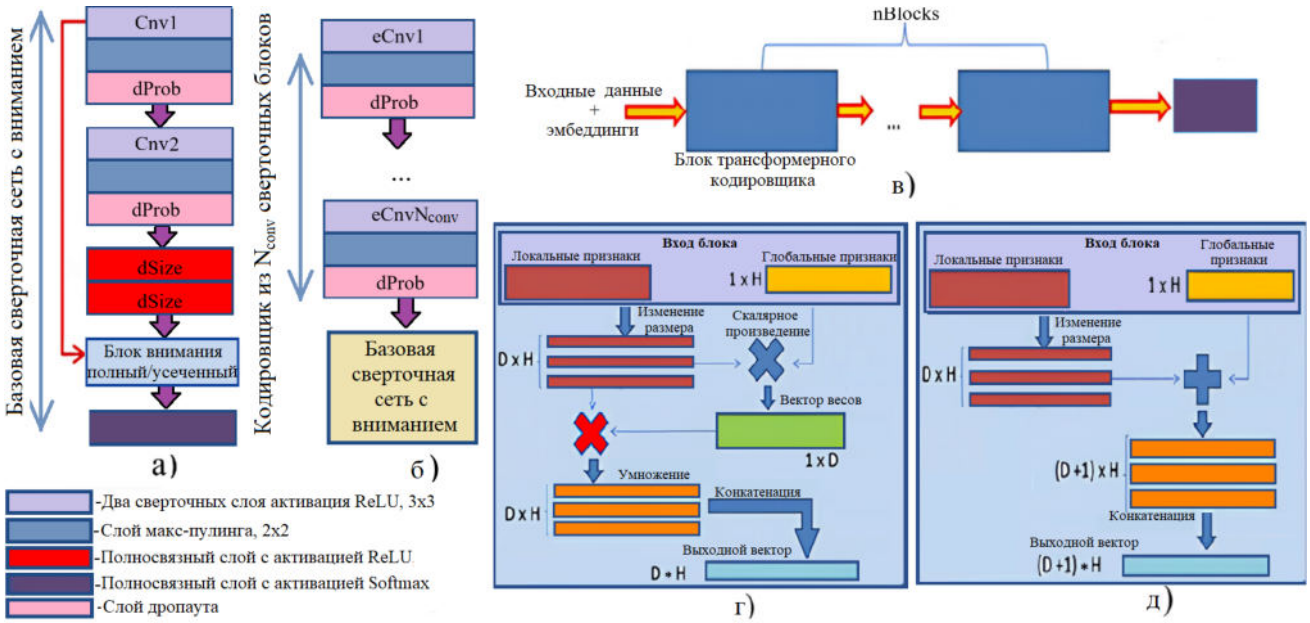

Рисунок 2.6 — Нейросетевые классификаторы фрагментов изображения: (а) «Малая» сверточныая сеть с двумя взаимозаменяемыми блоками внимания («полным» и «усеченным»); (б) «Большая» сверточная сеть с вниманием на основе скалярного произведения; (в) Трансформер; (г) блок внимания на основе скалярного произведения («полный»); (д) блок внимания на основе конкатенации («усеченный»)

$Cnv2$, слои макс-пулинга и дропаута с вероятностью исключения $dProb$. На рисунке 2.6 (б) «Малая сеть» становится глубже за счёт добавления N_{conv} сверточных блоков, и число каналов в каждом равняется $eConv_i, i = \overline{1,N_{conv}}$ («Полная сеть»). ViT используется для сравнения PrINN с популярными современными нейросетевыми архитектурами, которые можно применять для сегментации изображений по фрагментам. Он состоит из $nBlocks$ блоков кодировщика-трансформера.

«Полный» блок внимания [198] использует операцию скалярного произведения для определения уровня значимости каждого канала изображения, а «усеченный» блок – конкатенацию. Блоки внимания встраиваются в сверточную сеть перед последним полносвязным слоем, формирующем выходную метку класса фрагмента, и принимают на вход данные из предыдущих сверточных слоев. Объекты ближайшего к блоку сверточного слоя называются глобальными признаками размерности $1 \times N_{conv}$. Признаки из других слоёв считаются локальными. С помощью полносвязного слоя глобальные признаки преобразуются к размерности $1 \times H$, а локальные – к размерности $D \times H$. Затем в «полном» блоке внимания вычисляется скалярное произведение каж-

дого из D локальных векторов и вектора глобальных признаков, после чего к результату применяется функция $softmax(\cdot)$. Каждый элемент полученного вектора весов размерности $1 \times D$ умножается на один из D локальных векторов. Затем эти произведения объединяются в финальном полносвязном слое. Блок «усеченного» внимания пропускает оценку весов признаков: в нем локальные и глобальные признаки объединяются в общий выходной вектор.

Представленные на рисунке 2.6 сверточные и трансформерные архитектуры принимают на вход «расширенное» многоканальное изображение, каналы которого состоят из яркостей пикселей X_l и их вероятностей принадлежать к K компонентам смеси: $(X_l, X_l \cdot p_1^*(X_l), \ldots, X_l \cdot p_K^*(X_l))$. Чтобы эффективно использовать признаки смешанной модели, необходимо обеспечить сопоставимость интенсивности яркости пикселей и вероятностей. Для этого вероятности умножаются на яркость пикселей изображения, а затем эти значения нормализуются путём деления на 255. Если нормализовать только яркость, ее среднее будет меньше среднего значения элементов вектора вероятностей. Средняя яркость изображения составляет обычно 128, а количество пикселей с яркостью 255, которые после нормализации равняются 1, обычно невелико. Вероятности же, полученные с помощью EM алгоритма, нередко близки к 1, особенно для однородных участков. Из-за этого дисбаланса нейронная сеть уделяет больше внимания каналам расширенного изображения, содержащим значения вероятности, и игнорирует каналы со значениями яркости, что может привести к потере информации.

2.5 Результаты обработки радиолокационных изображений

2.5.1 Исследуемые данные

Архитектура PrINN была протестирована на семи радиолокационных изображениях, полученных различными по характеристикам радиолокаторами:

– Sentinel-1 [199]: спутниковый радиолокатор, рассмотрено три изображения разрешением 20 м;

– Capella [200]: спутниковый радиолокатор, рассмотрено одно изображение разрешением 0.5 м;
– ESAR [201]: авиационный радиолокатор, рассмотрено два изображения разрешением 2 м;
– датасет HRSID [202]: рассмотрено одно изображение разрешением 7 м.

На рисунке 2.7 представлены все исследуемые радиолокационные снимки с цветными контурами однородных областей различных поверхностей, числовые метки на которых означают номера классов (см. разделы 2.5.3–2.5.6).

Исследуемые снимки различаются как характеристиками радиолокаторов, так и типами подстилающих поверхностей. Изображения, полученные с помощью радиолокаторов ESAR и Capella, обладают более высоким уровнем зашумленности по сравнению со снимками данными Sentinel-1 и HRSID. На рисунке 2.8 представлены графики плотностей распределений классов для всех рассмотренных изображений, оцененные по эталонным площадкам. Высокая степень перекрытия областей под этими кривыми говорит о высоком уровне внутриклассовой дисперсии и, соответственно, сильной неоднородности исследуемых наборов.

При тестировании PrINN каждое изображение на рисунках 2.7 (а)–(ж) рассматривается как отдельный набор данных. Их предварительная обработка состоит из нескольких этапов. Затем выполняется ручная разметка изображений, поскольку тестовые изображения являлись либо неразмеченными, либо частично размеченными: например, на снимках HRSID были нанесены только контура кораблей. На каждом изображении цветными прямоугольниками (цвет соответствует метке класса) выделялись области, соответствующие только одному выбранному типу поверхности. Пример выделения обучающих областей для рисунка 2.7 (в) приведен на рисунке 2.9. Выделенные прямоугольники разбиваются на фрагменты размером 11×11 пикселей: использование более крупных фрагментов может противоречить предположению об однородности, согласно которому все его пиксели относятся к одному типу поверхности.

Также на каждом изображении были выделены тестовые однородные области, ограниченные контурами без заливки (см. рисунки 2.7 и 2.9). Пиксели этих областей, за исключением принадлежавших обучающей выборке, использовались для оценки точности сегментации.

В таблице 12 указано количество обучающих фрагментов для каждого изображения до и после аугментации на основе случайных сдвигов (от 0 до

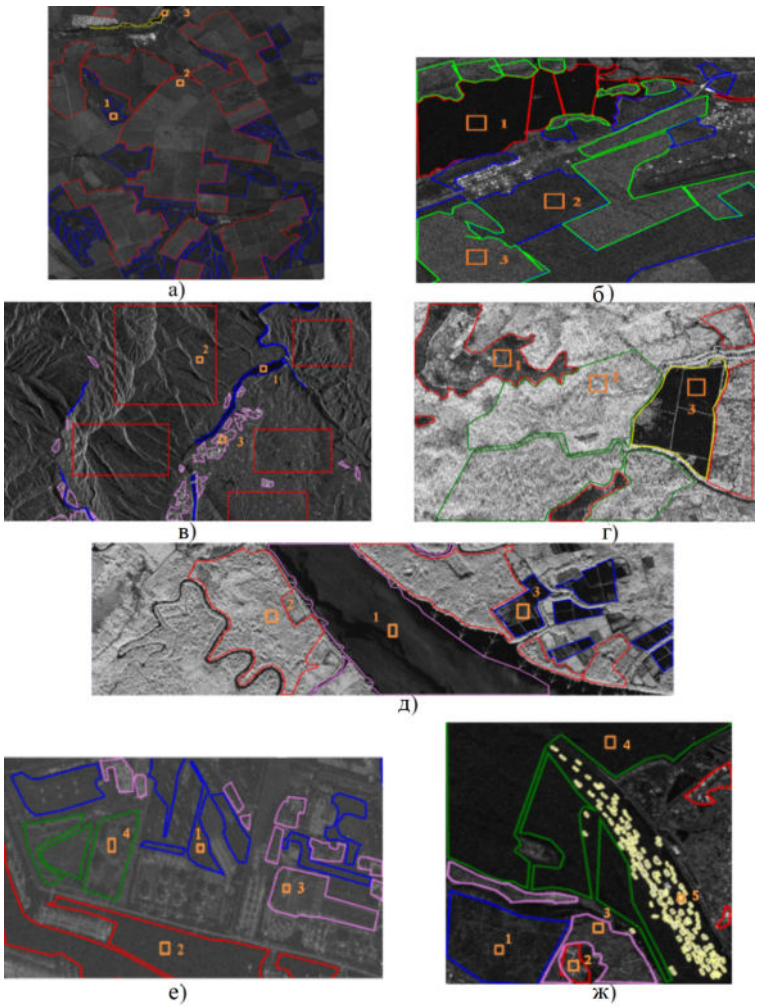

Рисунок 2.7 — Исследуемые радиолокационные изображения: Sentinel-1 (а, б и в), ESAR (г и д), Capella (е) и HRSID (ж)

половины длины фрагмента) и поворотов (от 0 до 180°). В среднем обучающая выборка для одного изображения содержит около 5500-6500 элементов. Доля пикселей, извлеченных для формирования обучающего набора, представлена в строке «Доля обучающих данных» таблицы 12. Во всех случаях тестовый набор намного больше и разнообразнее обучающего, что позволяет использовать его для оценки обобщающей способности модели и вероятности переобучения

Параметры смеси, за исключением весовых коэффициентов, определялись по аннотированным фрагментам [122]. Квадродерево состояло из четырех слоев, а размерность его нижнего слоя составляла 16×16 пикселей. Этого было достаточно для восстановления пространственных связей между пикселями при обработке по фрагментам.

По размеченным областям вычислялись значения метрик Precision (Prec), Recall (Rec) и F_1-меры (F_1) для каждого класса:

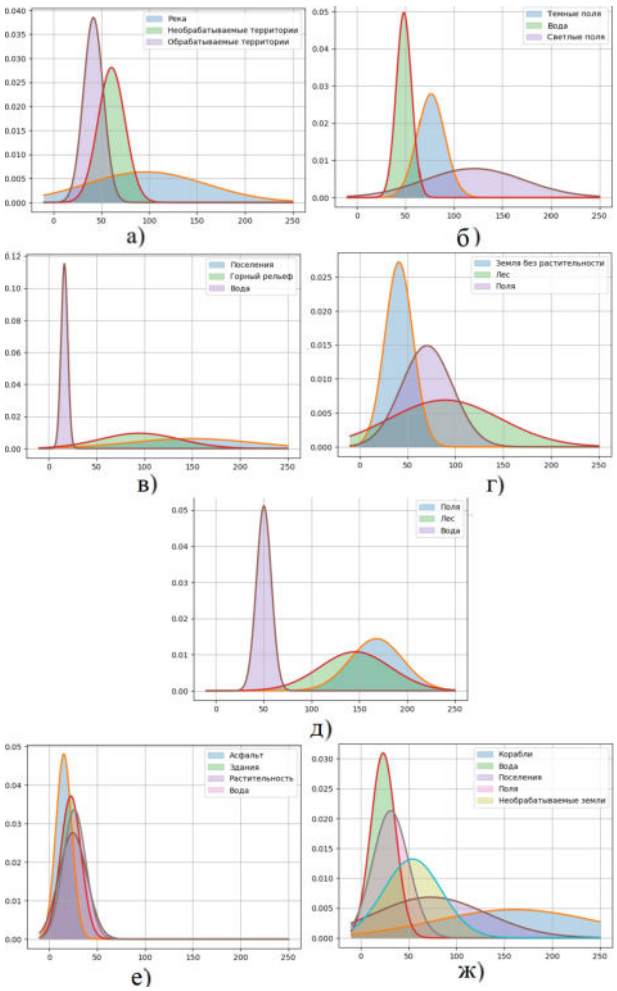

Рисунок 2.8 — Распределения классов рассмотренных изображений: Sentinel-1 (а, б и в), ESAR (г и д), Capella (е) и HRSID (ж)

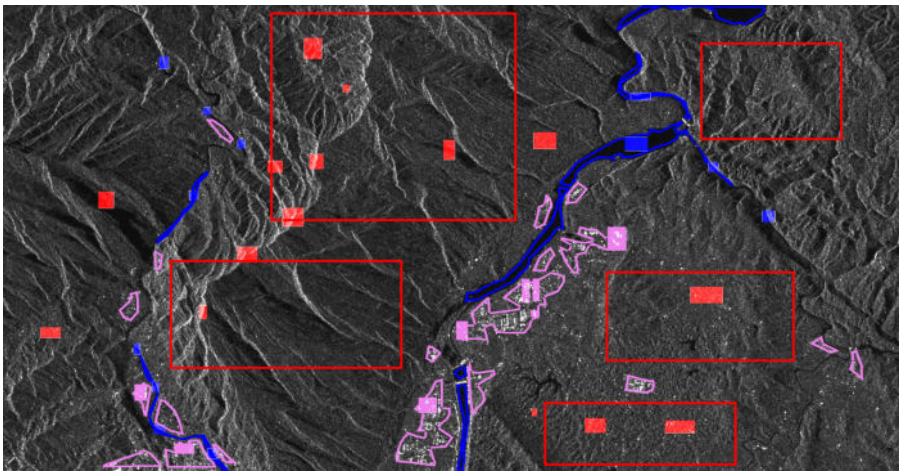

Рисунок 2.9 — Выделение обучающих фрагментов из снимка 2.7 (в). Области снимка, используемые для обучения сетей, отмечены цветными прямоугольниками (синими, красными, и фиолетовыми), а области, ограниченные цветными контурами без заливки, используются для оценки точности сегментации во время тестирования

Таблица 12 — Описание обучающих наборов выделенных из снимков, представленных на рисунках 2.7 (а)-(ж).

Количество патчей	**Sentinel-1 (Рис. 2.7а)**	**Sentinel-1 (Рис. 2.7б)**	**Sentinel-1 (Рис. 2.7в)**	**ESAR (Рис. 2.7г)**	**ESAR (Рис. 2.7д)**	**Capella (Рис. 2.7е)**	**HRSID (Рис. 2.7ж)**
Исходные данные	957	183	1102	3673	1338	919	343
Данные после аугментации	5742	1098	6612	22040	8032	5514	2058
Доля обучающих данных	0.97%	1.01%	1.7%	7.4%	4.0%	8.01%	2.9%

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN},$$
$$F_1 = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}, \tag{2.10}$$

где TP (истинно положительные) — это все пиксели целевого класса, которые были правильно классифицированы, FP (ложно положительные) — это пиксели, которые были неправильно классифицированы как пиксели целевого класса, а FN (ложно отрицательные) — это пиксели целевого класса, которые были классифицированы как пиксели другого класса. Стандартная метрика Accuracy, определяемая формулой (2.11), используется для оценки качества полной сегментации изображения и выбора оптимальной конфигурации нейронной сети:

$$Accuracy = \frac{True\ predictions}{All\ predictions}. \tag{2.11}$$

2.5.2 Гиперпараметры

В качестве базовых сегментаторов с которыми сравнивалась PrINN, рассматривались четыре типа сверточных и трансформерных архитектур. Далее

в работе приведены результаты для лучших по величине Accuracy конфигураций этих архитектур:

– Product Attention Small (PAS) – сверточная сеть из 6 слоев и блоком внимания на основе скалярного произведения (4 сверточных и 2 полносвязных слоя, см. рисунок 2.6 (а);
– Concatenated Attention Small (CAS) – сверточная сеть из 6 слоев и блоком внимания на основе конкатенации (см. рисунок 2.6 (а);

Таблица 13 — Описание гиперпараметров сетей

Параметр	**Описание**	**Архитектура**	**Диапазон**	**Лучшие значения**
dSize	Размерность двух выходных полносвязных слоев	PAS, CAS, PAF	15 – 169	15 (PAS, CAS), 60 (PAF)
Cnv1	Число каналов первого сверточного слоя в «Малой сети»	PAS, CAS, PAF	5 – 85	5 (PAS, CAS), 45 (PAF)
Cnv2	Число каналов второго сверточного слоя в «Малой сети»	PAS, CAS, PAF	5 – 85	15 (PAS, CAS), 65 (PAF)
dProb	Вероятность исключения элементов в сверточных слоях	PAS, CAS, PAF	0.15 – 0.3	0.25 (PAS, CAS, PAF)
N_{conv}	Число сверточных блоков в кодировщике	PAF	1 – 3	1
$eConv_i, i = 1 \dots N$	Число каналов в сверточных слоях кодировщика в «Полной сети»	PAF	5 – 85	25
nBlocks	Количество блоков кодировщика в трансформере	ViT	1 – 8	8

– Product Attention Full (PAF) – сверточная сеть из 10 слоев и блоком внимания на основе скалярного произведения (8 сверточных и 2 полносвязных слоя, см. рисунок 2.6 (б);
– Visual Transformer (ViT) –трансформерная архитектура, состоящая из 8 блоков трансформерного кодировщика (см. рисунок 2.6 (в).

Значения гиперпараметров, соответствующие конфигурациям PAS, CAS, PAF и ViT, их названия и диапазоны изменения представлены в таблице 13. Для их настройки использовался алгоритм случайного поиска [203].

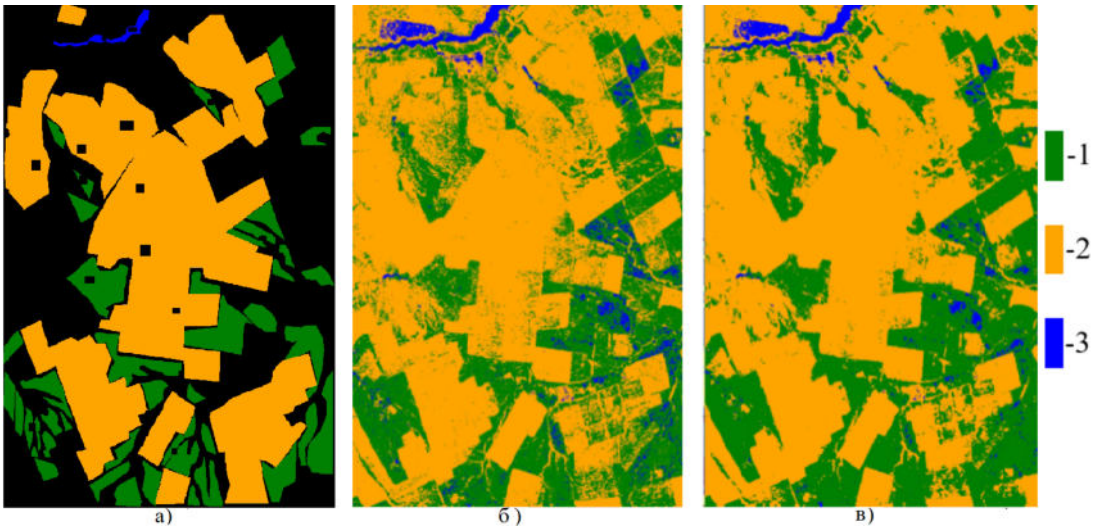

Рисунок 2.10 — Маски классов (а) изображения Sentinel-1 (рисунок 2.7а) и его сегментация с помощью сети PAF (б) и архитектурой PrINN на основе сети PAF (в). Цвет пикселей соответствует номеру класса: зеленый (1), оранжевый (2) и синий (3)

Таблица 14 — Результаты сегментации (значения метрик Recall, Precision, Accuracy ×100%) для снимка Sentinel-1 (рисунок 2.7 (а)).

НС	Класс 1				Класс 2				Класс 3				Accuracy	
	Rec		Prec		Rec		Prec		Rec		Prec			
	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN
PAS	**91.93**	89.70	98.04	96.29	90.11	83.97	77.27	74.59	78.92	81.17	**90.42**	22.35	**91.37**	88.15
CAS	89.50	86.51	**99.04**	98.44	88.30	81.7	79.36	68.71	88.98	90.68	53.78	16.93	89.19	85.28
PAF	89.15	90.24	98.55	96.92	89.32	85.57	**79.64**	77.45	81.10	83.76	81.84	35.24	89.14	88.98
ViT	88.95	87.32	98.34	98.23	**92.48**	80.60	77.12	72.63	81.39	**96.82**	87.921	15.45	89.82	85.62

Таблица 15 — Результаты сегментации (значения метрики $F_1 \times 100\%$) для снимка Sentinel-1 (рисунок 2.7 (а)).

НС	Класс 1		Класс 2		Класс 3	
	PrINN	vNN	PrINN	vNN	PrINN	vNN
PAS	**94.89**	90.84	84.28	51.33	**86.14**	77.33
CAS	94.03	92.09	67.05	28.53	83.59	74.64
PAF	93.61	93.35	81.46	20.81	84.20	75.90
ViT	93.41	92.45	**84.53**	26.65	84.11	76.41

2.5.3 Изображения Sentinel-1

На рисунке 2.7 (а) выделенны три класса: необрабатываемые территории (класс 1), обрабатываемые земли (класс 2) и река (класс 3). На этом снимке класс 3 соответствуем категории небольших объектов. В таблицах 14 и 15 представлены максимальные значения Precision, Recall, F_1-меры и Accuracy для первого снимка Sentinel-1, обработанного либо нейронной сетью без информирования (столбец «vanilla Neural Network» или «vNN») или PrINN на основе этой архитектуры. Лучшие результаты выделены жирным шрифтом.

Таблица 16 — Результаты сегментации (значения метрик Recall, Precision, Accuracy $\times 100\%$) для снимка Sentinel-1 (рисунок 2.7 (б)).

НС	Класс 1				Класс 2				Класс 3				Accuracy	
	Rec		Prec		Rec		Prec		Rec		Prec			
	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN
PAS	94.4	94.0	**100**	86.0	93.4	82.9	94.0	84.0	94.8	84.0	92.0	90.0	94.2	86.2
CAS	**99.7**	99.3	99.0	98.0	93.3	93.7	97.0	94.0	97.0	92.0	94.0	95.0	**97.5**	94.6
PAF	99.6	98.5	98.0	98.0	94.8	**95.9**	97.0	75.0	96.4	69.0	**96.0**	95.0	97.2	86.5
ViT	99.5	–	99.0	–	89.0	–	**98.0**	–	**98.9**	–	79.0	–	95.3	–

Для всех архитектур значение Recall класса 3 снижается, в то время как Precision увеличивается более чем на 30–50%. При обработке изображения по фрагментам границы объектов могут систематически искажаться. Для небольших объектов искажение небольшого количества пикселей может привести к значительным изменениям метрик, например к снижению Recall для класса 3. Однако за счет уменьшения ложно положительных результатов (частоты ошибок первого рода, ошибочных отклонений нулевой гипотезы) точность выделения класса 3 повышается. Для классов 1 и 2, соответствующих крупным объектам, при обработке PrINN увеличиваются как Recall, так и Precision. Таким образом, использование композиции вероятностных моделей повышает качество обработки изображения. PrINN демонстрирует наилучшие по точности значения для всех архитектур – лучший результат получен с помощью PrINN на основе PAS – на рисунке 2.10 сравниваются соответствующие результаты сегментации.

На рисунке 2.7 (б) выделено три типа поверхностей: вода (класс 1), тёмные поля (класс 2), светлые поля (класс 3). Хотя и первый и второй снимки Sentinel-1 изображают схожие типы поверхностей и объектов, их отражающие свойства значительно отличаются. Внутриклассовые различия для пикселей второго изображения менее выражены, чем для первого (см. рисунок 2.8). Максимальная классовая дисперсия для первого изображения составляет 60, для второго — 20.

В таблицах 16 и 17 представлены значения Recall, Precision, Accuracy и F_1-меры для второго изображения Sentinel-1. Набор данных слишком мал для обучения архитектуры ViT без информирования: сеть систематически не

Таблица 17 — Результаты сегментации (значения метрики $F_1 \times 100\%$) для снимка Sentinel-1 (рисунок 2.7 (б)).

НС	Класс 1		Класс 2		Класс 3	
	PrINN	vNN	PrINN	vNN	PrINN	vNN
PAS	97.11	89.82	93.69	83.44	93.37	86.89
CAS	**99.35**	98.64	95.11	93.84	95.47	93.47
PAF	98.79	98.24	**95.88**	84.17	**96.19**	79.93
ViT	99.25	–	93.28	–	87.83	–

Таблица 18 — Результаты сегментации (значения метрик Recall, Precision, Accuracy ×100%) для снимка Sentinel-1 (рисунок 2.7 (в)).

НС	Класс 1				Класс 2				Класс 3				Accuracy	
	Rec		Prec		Rec		Prec		Rec		Prec			
	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN
PAS	59.97	62.34	**78.62**	52.62	**98.76**	95.89	90.71	89.23	27.32	11.24	87.55	53.40	90.15	86.31
CAS	81.29	82.88	72.42	43.65	97.84	92.24	92.94	92.90	39.87	37.38	89.20	65.52	**91.61**	86.67
PAF	69.92	69.56	62.44	54.35	96.38	94.85	93.01	91.84	**46.41**	33.91	77.30	62.42	90.39	87.90
ViT	**89.36**	87.47	41.73	36.72	91.53	87.65	**93.14**	93.16	34.83	39.93	53.92	42.45	86.55	83.53

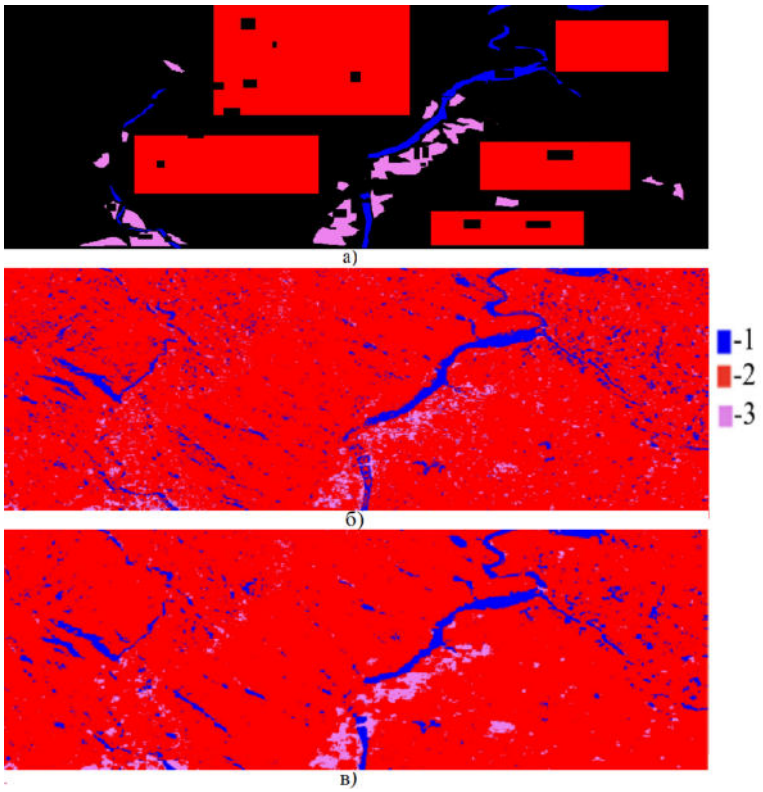

Рисунок 2.11 — Маски классов (а) изображения Sentinel-1 (рисунок 2.7 (в) и его сегментация с помощью сети PAF (б) и архитектурой PrINN на основе PAF (в). Цвет пикселей соответствует номеру класса: синий (1), красный (2) и фиолетовый (3)

различает классы рассмотренных поверхностей. Поэтому в соответствующих позициях таблиц стоят прочерки. Наилучшие значения Accuracy были получены с использованием PrINN на основе CAS.

Таблица 19 — Результаты сегментации (значения метрики $F_1 \times 100\%$) для снимка Sentinel-1 (рисунок 2.7 (в)).

НС	Класс 1		Класс 2		Класс 3	
	PrINN	vNN	PrINN	vNN	PrINN	vNN
PAS	68.04	57.06	94.56	92.44	41.64	18.57
CAS	**76.59**	57.18	**95.33**	92.57	55.11	47.60
PAF	65.97	61.02	94.66	93.32	**57.99**	43.95
ViT	56.89	51.72	92.33	90.32	42.32	41.15

На рисунке 2.7 (в) изображены русла рек (класс 1), сильно неоднородный горный рельеф (класс 2) и несколько населённых пунктов (класс 3). Горный рельеф занимает большую часть изображения. Чтобы оценить точность его обработки, было выделено несколько областей (см. рисунок 2.7 (в), красные контуры), соответствующих разным типам рельефа: горным хребтам, вершинам или плато. Результаты сегментации представлены на рисунке 2.11 и в таблицах 18 и 19. Наилучшие значения точности были получены при использовании PrINN на основе CAS.

Значение метрики Recall для классов 2 и 3 повышается при обработке с помощью PrINN. Для класса 1, представленного небольшими объектам, в некоторых случаях Recall снижается, но при этом значение его Precision увеличивается на 8.08-28.77% (среднее – 18.87%, медиана — 26.0%). Это свидетельствует об улучшении качества классификации пикселей и снижении вероятности ошибки первого рода для класса 1.

Для изображения с рисунка 2.7 (в) дополнительно была выполнена перекрестная проверка результатов. Была использована специальная стратегия для разделения обучающего набора на части. Набор вырезанных фрагментов был разделен на 3 подгруппы в соответствии с номером класса. Затем каждая была разделена на 5 частей одинакового размера: для перекрестной проверки одна из них удалялась из подгруппы. После этого каждое подмножество классов имело только 80% от своего первоначального объема. PrINN и сети без информирования обучались на пяти таких сокращенных наборах. Среднее значение F_1-меры и Accuracy, полученные в ходе сегментации всех сокращенных наборов для рисунка 2.7 (в) представлены в таблице 20. Максимальные и минимальные

Таблица 20 — Результаты перекрестной проверки (5 диапазонов) (средние значения и диапазоны изменения метрик F_1, $Accuracy \times 100\%$) для снимка Sentinel-1 (рисунок 2.7 (в))

НС	Класс 1		Класс 2		Класс 3		Accuracy	
	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN
PAS	**72.03 (68.80-74.51)**	53.81 (47.22-60.66)	**94.03 (93.56-94.49)**	91.20 (90.0-93.06)	28.14 (18.65-43.27)	**34.71 (21.53-41.25)**	**89.72 (88.80-90.60)**	84.99 (83.29-87.99)
CAS	**59.55 (50.75-63.33)**	55.56 (46.54-71.4)	**93.20 (91.32-94.28)**	91.27 (89.46-93.44)	50.81 (45.67-**54.59**)	**50.92** (50.86-51.05)	**88.05 (84.87-89.93)**	85.29 (82.54-88.87)
PAF	**58.97 (49.74-65.03)**	52.10 (46.25-57.00)	**93.11 (91.49-93.80)**	92.05 (91.62-92.31)	52.23 (48.99-53.85)	**53.87 (52.17-55.56)**	**87.86 (85.39-88.97)**	86.41 (85.55-86.89)
ViT	**56.58 (49.33-67.54)**	53.44 (43.87-68.37)	**92.45 (91.37-93.61)**	90.59 (89.80-91.63)	**44.31 (42.36-48.20)**	43.96 (42.92-45.69)	**86.74 (84.91-88.88)**	84.17 (82.82-85.66)

значения показателей указаны в квадратных скобках, а наилучшие результаты выделены жирным шрифтом.

PrINN демонстрирует наилучшие показатели Accuracy во всех случаях. Увеличение средних значений составило 4.73%. Для большинства классов значения F_1-меры, полученные PrINN, также выше, чем у сетей без информирования: прирост средних значений достигает 18.22%. Более того, результаты, полученные при обучении на сокращенном наборе данных, близки к результатам, полученным на полном: разница максимальных значений Accuracy не превосходит 2.33%.

Показатели точности сегментации для класса 3 (поселения) немного снижаются при обработке с помощью PrINN по сравнению с результатами обычных сетей из-за специфики обработки изображений по фрагментам (см. также раздел 2.5.3). Это можно увидеть на примере сети CAS в таблице 20. Однако общие показатели улучшаются, что выражается в значительном приросте F_1-меры для остальных классов и общему увеличению Accuracy по сравнению с сетями без информирования.

Таким образом, при информировании композицией вероятностных моделей показатели точности сегментации изображений Sentinel-1 улучшаются по сравнению с результатами обычных архитектур. Прирост точности является стабильным и заметным на разных конфигурациях обучающего набора.

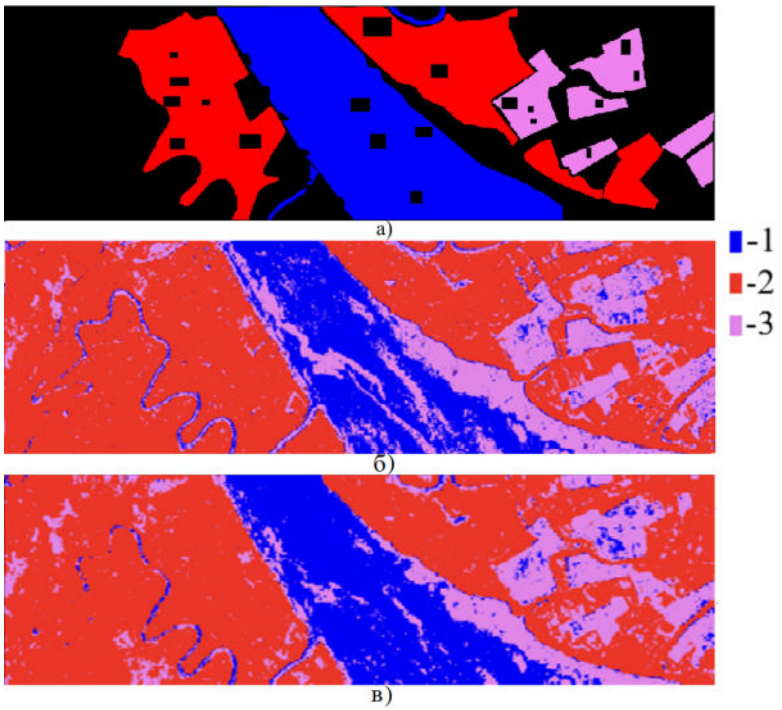

Рисунок 2.12 — Маски классов (а) изображения ESAR (рисунок 2.7 (д) и его сегментация с помощью сети PAS (б) и архитектурой PrINN на основе сети PAS (в). Цвет пикселей соответствует номеру класса: синий (1), красный (2) и фиолетовый (3)

Таблица 21 — Результаты сегментации (значения метрик Recall, Precision, Accuracy ×100%) для снимка ESAR (рисунок 2.7 (г))

НС	**Класс 1**				**Класс 2**				**Класс 3**				**Accuracy**	
	Rec		Prec		Rec		Prec		Rec		Prec			
	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN
PAS	79.5	70.0	63.0	56.0	98.8	92.0	82.0	77.0	35.1	27.3	79.0	63.0	73.9	66.3
CAS	**82.9**	72.0	72.0	69.0	**99.9**	98.0	86.0	80.0	54.0	48.2	**85.0**	76.0	80.9	75.9
PAF	80.4	75.0	74.0	68.0	98.9	95.8	89.0	86.0	62.7	54.6	84.0	77.0	82.9	77.7
ViT	65.0	58.0	**85.0**	76.0	97.6	93.8	**91.0**	86.0	**79.4**	66.9	76.0	69.0	**84.0**	76.7

Таблица 22 — Результаты сегментации (значения метрики $F_1 \times 100\%$) для снимка ESAR (рисунок 2.7 (г)).

НС	Класс 1		Класс 2		Класс 3	
	PrINN	vNN	PrINN	vNN	PrINN	vNN
PAS	70.29	62.22	89.62	83.83	48.60	38.09
CAS	77.06	70.46	92.43	88.08	66.04	58.98
PAF	**77.07**	71.33	93.68	90.63	71.80	63.89
ViT	73.66	65.79	**94.18**	89.73	**77.66**	67.93

Таблица 23 — Результаты сегментации (значения метрик Recall, Precision, Accuracy $\times 100\%$) для снимка ESAR (рисунок 2.7 (д))

НС	Класс 1				Класс 2				Класс 3				Accuracy	
	Rec		Prec		Rec		Prec		Rec		Prec			
	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN
PAS	**88.32**	79.35	96.11	78.87	**95.92**	95.86	98.24	97.39	83.71	25.87	**61.74**	24.96	**91.21**	80.16
CAS	85.59	91.21	**96.47**	82.32	78.78	94.53	99.22	98.70	**89.12**	37.73	41.97	47.56	82.93	86.01
PAF	87.28	72.44	95.85	94.97	95.75	94.82	97.69	97.68	82.99	83.59	60.58	44.11	90.60	84.05
ViT	83.87	–	92.90	–	96.51	–	**99.31**	–	83.75	–	58.71	–	89.62	–

2.5.4 Изображения ESAR

Оба изображения, полученные радиолокатором ESAR (см. рисунки 2.7 (г) и (д)), содержат похожие объекты: леса и возделанные поля рядом с ними. На рисунке 2.7 (г) изображены участки земли без растительности (класс 1), лес (класс 2) и возделанные поля (класс 3). Значения Recall, Precision и F_1-меры, полученные при обработке первого снимка ESAR, представлены в таблицах 21 и 22. Максимальные значения для каждого класса выделены жирным шрифтом: лучшие значения метрики Accuracy были получены с использованием PrINN на основе ViT.

Таблица 24 — Результаты сегментации (значения метрики $F_1 \times 100\%$) для снимка ESAR (рисунок 2.7 (д))

НС	Класс 1		Класс 2		Класс 3	
	PrINN	vNN	PrINN	vNN	PrINN	vNN
PAS	**92.05**	79.11	97.07	96.61	71.07	25.40
CAS	90.70	86.54	87.83	96.57	57.06	42.07
PAF	91.36	82.18	96.71	96.22	70.03	57.74
ViT	88.15	–	**97.89**	–	**69.03**	–

На рисунке 2.7 (д) представлены 3 типа поверхностей: вода (класс 1), лес (класс 2) и возделанные поля (класс 3). Это радиолокационное изображение является зашумленным, и объекты на нем характеризуются выраженной внутриклассовой изменчивостью. При этом обучающий набор данных сильно ограничен, что приводит к систематическим ошибкам сегментации сетями CAS и ViT без информирования. Значения Recall и Precision, полученные с помощью этих двух сетей, не использовались для оценки средних и медианных показателей точности. На рисунке 2.12 продемонстрированы результаты сегментации сетью PAS без информирования и PrINN на основе PAS. Значения Recall, Precision и F_1-меры для каждой архитектуры, полученные для изображения с рисунка 2.7д), представлены в таблицах 23 и 24. Лучшие значения Accuracy получены с помощью PrINN на основе PAS.

PrINN повышает точность сегментации изображений ESAR – для двух изображений прирост F_1-меры достигает 45.65%, а Accuracy – 11.05% в сравнении с результатами, полученными архитектурами без информирования.

2.5.5 Изображения Capella

На снимке, полученном с помощью радиолокатора Capella (см. рисунок 2.7 (е)), представлен городской ландшафт, включающий 4 типа поверхностей: асфальт (класс 1), вода (класс 2), здания (класс 3) и зелёные насаждения (класс 4). Радиолокатор Capella имеет очень высокое разрешение – 0.5 м на пиксель.

Таблица 25 — Результаты сегментации (значения метрик Recall, Accuracy ×100%) для снимка Capella (рисунок 2.7 (e))

НС	Класс 1		Класс 2		Класс 3		Класс 4		Accuracy	
	Rec		Rec		Rec		Rec			
	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN
PAS	70.61	–	52.45	–	**88.73**	–	45.58	–	68.27	–
CAS	**73.58**	42.43	61.16	**86.58**	87.47	68.12	38.18	37.61	**69.73**	58.11
PAF	62.41	52.56	63.96	77.18	87.55	58.64	34.04	28.63	65.65	55.74
ViT	64.52	37.64	65.97	79.08	83.17	74.94	34.47	**63.82**	65.67	60.55

Таблица 26 — Результаты сегментации (значения метрик Precision, ×100%) для снимка Capella (рисунок 2.7 (e))

НС	Класс 1		Класс 2		Класс 3		Класс 4	
	Prec		Prec		Prec		Prec	
	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN
PAS	71.06	–	**89.27**	–	59.94	–	67.08	–
CAS	**76.36**	60.47	79.87	46.08	58.91	**74.53**	77.45	56.89
PAF	73.02	64.83	69.84	52.34	57.38	58.59	71.99	35.26
ViT	70.99	70.17	62.35	49.13	60.52	63.87	**80.94**	66.57

Это одновременно повышает как детальность, так и зашумленность изображения, особенно когда рельеф подстилающей поверхности неоднороден и имеет много перепадов высот. В результате межклассовые различия в яркости пикселей не столь значительны, о чём свидетельствует сходство их вероятностных распределений. Среднее значение для класса 1 составляет 67.39, а дисперсия — 7.74. Среднее значение для класса 2 составляет 71.81, а дисперсия — 6.7. Среднее значение для класса 3 составляет 64.9, а дисперсия — 9.44. Из-за высокого уровня шума снимок Capella сложно эффективно сегментировать: в сравнении с другими изображениями полученные для него значения метрик существенно ниже. Из-за сходства классов сеть PAS без информирования систематически не

Таблица 27 — Результаты сегментации (значения метрики $F_1 \times 100\%$) для снимка Capella (рисунок 2.7 (е))

НС	Класс 1		Класс 2		Класс 3		Класс 4	
	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN
PAS	70.83	–	66.07	–	71.54	–	50.61	–
CAS	**74.94**	49.86	**69.27**	60.14	70.40	71.18	51.14	45.28
PAF	67.29	57.81	66.77	62.38	69.32	58.61	46.22	31.60
ViT	58.93	62.21	67.58	62.96	**75.03**	68.09	**74.47**	45.70

различает и теряет некоторые классы на снимке: показатели Recall, Precision и F_1-меры для этой архитектуры не приведены в таблицах 25, 26 и 27. Максимальное значение Accuracy получено PrINN на основе CAS. Результаты сегментации представлены на рисунке 2.13.

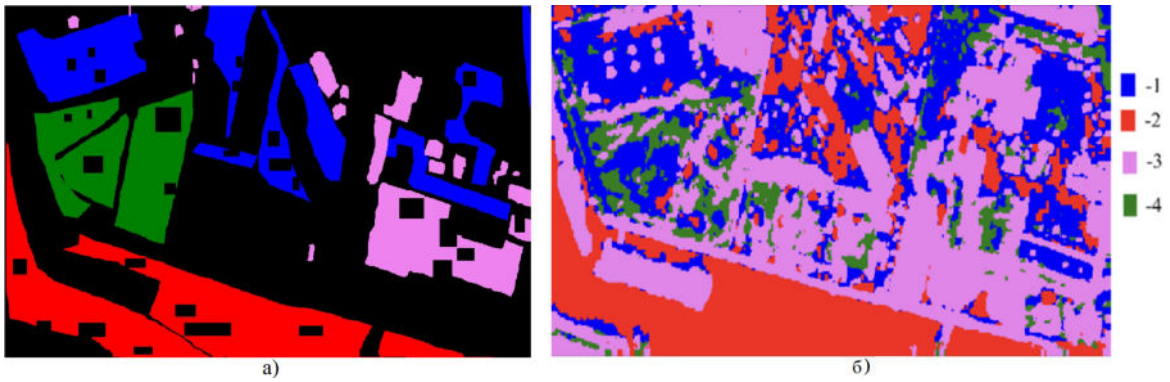

Рисунок 2.13 — Маски классов (а) изображения Capella (рисунок 2.7 (е)) и его сегментация с помощью PrINN на основе сети CAS (б). Цвет пикселей соответствует номеру класса: синий (1), красный (2), фиолетовый (3) и зеленый (4)

Несмотря на значительное количество ошибок, возникающих при сегментации изображения Capella, при использовании PrINN показатели Recall, Precision и F_1-меры увеличиваются по сравнению с показателями, полученными при использовании сетей без информирования. Среднее значение F_1-меры увеличивается на 12.19%, а Accuracy – на 11.62%.

Таблица 28 — Результаты сегментации (значения метрик Recall и Accuracy ×100%) для снимка HRSID (рисунок 2.7 (ж))

НС	**Класс 1**		**Класс 2**		**Класс 3**		**Класс 4**		**Класс 5**		**Accuracy**	
	Rec		Rec		Rec		Rec		Rec			
	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN
PAS	93.4	69.3	63.1	40.5	**72.0**	52.3	94.8	75.6	70.0	14.4	90.2	70.1
CAS	**95.9**	83.3	**68.9**	44.2	61.1	66.5	**95.9**	93.9	**93.3**	23.6	**90.5**	86.5
PAF	93.2	80.9	61.9	49.7	70.3	50.12	94.5	91.4	81.8	38.7	89.9	82.7
ViT	94.2	–	68.1	–	61.8	–	93.4	–	93.1	–	88.6	–

Таблица 29 — Результаты сегментации (значения метрики Precision ×100%) для снимка HRSID (рисунок 2.7 (ж))

НС	**Класс 1**		**Класс 2**		**Класс 3**		**Класс 4**		**Класс 5**	
	Prec		Prec		Prec		Prec		Prec	
	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN
PAS	78.7	39.6	57.5	29.0	81.6	50.2	98.9	97.9	**10.3**	3.50
CAS	**79.9**	68.9	56.8	**68.8**	**82.3**	63.6	99.2	97.7	8.92	9.43
PAF	77.6	59.9	57.4	43.16	79.5	68.3	99.1	96.7	10.1	9.85
ViT	79.5	–	50.3	–	76.7	–	**99.6**	–	05.09	–

2.5.6 Изображения HRSID

На снимке из набора HRSID (см. рисунок 2.7 (ж)) изображены обработанные поля (класс 1), населенные пункты (класс 2), необработанные поля (класс 3), водная поверхность (класс 4) и корабли (класс 5). Значения Recall, Precision и Accuracy для каждой архитектуры представлены в таблицах 28 и 29, а значения F_1-меры – в таблице 30 (обучающий набор слишком мал для архитектуры ViT без информирования, поэтому в соответствующих позициях таблиц стоят прочерки). На рисунке 2.14 показана результаты сегментации изображения

Таблица 30 — Результаты сегментации (значения метрики $F_1 \times 100\%$) для снимка HRSID (рисунок 2.7 (ж))

НС	Класс 1		Класс 2		Класс 3		Класс 4		Класс 5	
	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN	PrINN	vNN
PAS	85.45	50.41	60.15	33.83	**76.53**	51.24	**96.83**	85.29	17.93	5.62
CAS	**87.15**	75.45	**62.24**	53.79	70.11	65.03	97.55	95.77	16.28	13.47
PAF	84.66	68.86	59.57	46.19	74.59	57.81	96.71	93.95	**17.98**	15.70
ViT	86.23	–	57.88	–	68.44	–	96.37	–	9.65	–

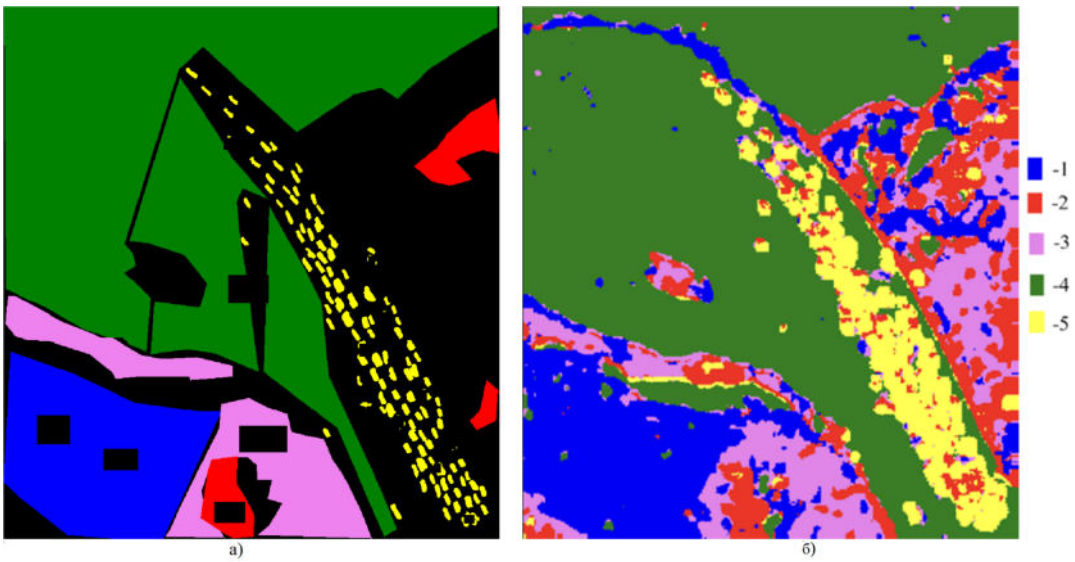

Рисунок 2.14 — Маски классов (а) изображения HRSID (рисунок 2.7 (ж)) и его сегментация с помощью PrINN на основе сети CAS (б). Цвет пикселей соответствует номеру класса: синий (1), красный (2), фиолетовый (3), зеленый (4) и желтый (5)

HRSID – лучшее значение Accuracy получено PrINN на основе CAS. Точность сегментации изображения из набора HRSID повышается при использовании PrINN. Увеличение F_1-меры для каждого класса достигает 35%, а Accuracy для полного изображения – до 20.3% в сравнении с результатами сетей без информирования.

Таблица 31 — Приросты метрик точности сегментации (диапазоны изменения и средние/медианные значения) для PrINN, FNM + vNN and vNN + QTree

Снимок	**PrINN**		**FNM + vNN**		**vNN + QTree**	
	F_1	Acc	F_1	Acc	F_1	Acc
Sentinel-1 (рис. 2.7 (а))	**19.24/ 8.55 (0.26-60.65)**	**3.82/ 3.93 (0.16-4.2)**	18.14/ 7.79 (0.5-56.59)	2.88/ 2.94 (1.54-4.15)	1.93/ 1.35 (0.19-6.01)	2.52/ 2.5 (1.08-4.01)
Sentinel-1 (рис. 2.7 (б))	**6.27/ 7.29 (0.54-16.26)**	**7.2/ 8.0 (2.9-10.7)**	5.60/ 6.45 (0.0-15.25)	5.56/ 6.9 (0.5-9.30)	3.25/ 2.76 (0.6-6.68)	3.3/ 3.0 (1.6-5.4)
Sentinel-1 (рис. 2.7 (в))	**13.00/ 9.24 (1.34-47.26)**	**4.7/ 4.93 (2.49-7.56)**	13.78/ 6.97 (0.97-47.03)	2.3/ 2.28 (1.91-2.75)	4.94/ 6.17 (1.78-10.06)	2.88/ 3.39 (1.4-3.65)
ESAR (рис. 2.7 (г))	**6.75/ 7.23 (3.05-10.51)**	**6.27/ 6.25 (5.0-7.6)**	2.41/ 1.92 (0.46-7.68)	2.03/2.0 (0.7-3.4)	5.89/ 6.07 (1.73-10.79)	5.4/ 5.8 (3.6-7.2)
ESAR (рис. 2.7 (д))	**12.19/ 9.57 (0.45-45.65)**	**8.79/ 11.04 (6.55-11.05)**	9.73/ 4.04 (0.0-34.00)	4.48/ 7.9 (1.07-8.01)	2.38/ 2.32 (0.73-4.41)	2.07/ 1.89 (1.53-2.53)
Capella (рис. 2.7 (е))	**12.19/ 9.12 (4.39-28.76)**	**8.88/ 9.91 (2.1-11.62)**	10.11/ 9.11 (0.28-22.90)	7.05/ 7.99 (7.9-11.1)	4.48/ 4.92 (0.20-8.66)	4.7/ 5.36 (0.94-7.81)
HRSID (рис. 2.7 (ж))	**12.74/ 13.38 (1.77-35.04)**	**10.48/ 7.14 (3.99-20.3)**	9.58/ 10.06 (0.37-30.46)	8.02/ 4.52 (0.7-10.58)	5.22/ 6.71 (0.51-16.25)	5.22/ 4.26 (0.87-8.52)

2.5.7 Сравнение с альтернативными подходами к информированию моделями смеси и квадродерева

В разделе представлено сравнение PrINN с альтернативными архитектурами, информированными конечной нормальной смесью (FNM) и квадродеревом (QTree). Во всех предшествовавших разделах PrINN сравнивалась с базовыми сегментаторами, которые применялись к изображениям без какой-либо последующей или предварительной обработки. В сравнении с ними PrINN демонстрирует увеличение всех показателей точности за счет использования композиции двух вероятностных моделей. Оценки вклада каждой модели в общий результат представлены в таблице 31. Столбец «FNM+vNN» соответствует обучению базового сегментатора на фрагментах, предобработанных смесью, т.е. с применением PrINN без обработки квадродеревом. Столбец «vNN+QTree»

соответствует обработке результатов базовых сегментаторов с помощью квадродерева. Каждая ячейка содержит среднее и медианное улучшение результатов базовых сегментаторов (лучшие показатели выделены жирным шрифтом). В скобках указаны диапазоны прироста значений метрик.

Согласно полученным результатам, использование каждой вероятностной модели по отдельности для информирования нейронных сетей улучшает сегментацию изображений, причем сети, информированные смесью, в большинстве случаев демонстрируют более высокие приросты точности относительно базовых сегментаторов, чем сети, информированные моделью квадродерева. Для «FNM+vNN» прирост среднего значения Accuracy достигает 8.02%, а среднего значения F_1-меры – до 18.14%, в то время как для «vNN+QTree» прирост среднего значения Accuracy достигает 5.4%, а среднего значения F_1-меры – до 5.89%. Однако использование композиции обеих моделей в PrINN дает более высокие по точности результаты по сравнению как с «FNM+vNN», так и с «vNN+QTree»: прирост среднего значения Accuracy достигает 10.38%, а среднего значения F_1-меры – до 19.24%. Используя композицию двух вероятностных моделей, архитектура PrINN использует больше дополнительной информации о яркостных и пространственных свойствах пикселей нежели сети, информированные только одной вероятностной моделью, что положительно сказывается на результатах сегментации изображений.

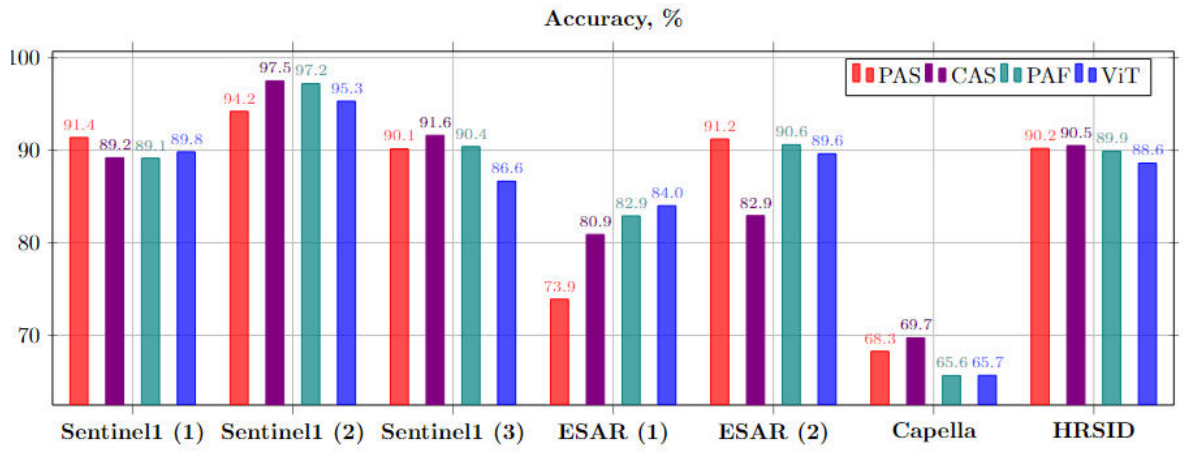

Рисунок 2.15 — Значения Accuracy, полученные архитектурой PrINN на основе архитектуры PAS (красный), CAS (фиолетовый), PAF (бирюзовый) и ViT (синий)

Значения Accuracy, полученные PrINN для каждого радиолокационного изображения, представлены на рисунке 2.15. В четырёх из семи случаев мак-

симальные значения Accuracy были получены при использовании PrINN на основе CAS. Повышение точности зависит неоднородности набора, объёма обучающей выборки и баланса классов. Более сложные архитектуры, такие как сети PAF и ViT, обычно плохо обучаются на небольших наборах. Однако при использовании подхода PrINN эти архитектуры могут быть успешно применены к обработке изображений. При этом чем меньше объём набора данных, тем более близкие результаты демонстрируют сложные и простые архитектуры нейронных сетей. Для 5 из 7 изображений размер обучающей выборки составлял менее 6000 фрагментов, а разница в значениях Accuracy между сложными и простыми архитектурами составляла всего 1.94%. По мере увеличения размера обучающей выборки более сложные архитектуры превосходят по точности более простые (см. результаты для снимков ESAR).

2.6 Выводы

В главе представлен метод информирования нейронных сетей композицией моделей конечной смеси вероятностных распределений и поля Маркова в форме квадродерева для повышения точности сегментации неоднородных наборов изображений. Доказана теорема 5 о повышении точности при информировании НС смешанной моделью. Доказывается теорема 6 о связи между байесовским алгоритмом вычисления вероятностей классов узлов в структуре квадродерева и обработки их графово-сверточной нейронной сетью. Теоремы 5

Таблица 32 — Оценки производительности PrINN и архитектур без информирования

НС	**Параметры** ($\times 10^3$)		**FLOPS** ($\times 10^6$)		**Длительность эпохи (с)**	
	PrINN	vNN	PrINN	vNN	PrINN	vNN
PAS	10.667	10.622	0.35274	0.33753	13.23	11.24
CAS	11.174	11.129	0.35119	0.33598	12.45	10.10
PAF	11.174	11.129	0.35119	0.33598	11.14	11.77
ViT	14 189	14 188	63.16	63.13	29.25	29.23

и 6 определяют способ реализации информирования: на уровне входных признаков для смешанной модели, и на уровне архитектуры – для квадродерева. Кроме того, доказывается свойство эргодичности поля Маркова в виде квадродерева (теорема 7).

Разработанная архитектура PrINN была протестирована для сегментации семи радиолокационных изображений, полученных с помощью различных радиолокаторов. В сравнении с рассмотренными сверточными и трансформерными архитектурами PrINN повышает точность сегментации всех снимков до 20.31% по метрике Accuracy и до 19.24% по метрике F_1.

PrINN сегментирует изображения с более высокой точностью, чем архитектуры, информированные каждой вероятностной моделью (смесью или квадродеревом) по отдельности. Информирование композицией моделей позволяет одновременно учитывать различные вероятностные свойства пикселей, что повышает максимальные и средние значения Recall, Precision, F_1-меры и Accuracy, см. раздел 2.5, что свидетельствует об эффективности представленного в главе подхода информирования композицией вероятностных моделей.

PrINN и другие рассмотренные архитектуры обучались на AMD Ryzen 5 5600H. Каждая эпоха обучения занимала не более 4 секунд. Для сегментации с помощью сетей без информирования было достаточно одного графического процессора NVIDIA V100. Количество параметров (в тысячах), число FLOPS (в миллионах) и время выполнения одной эпохи обучения (в секундах) представлены в таблице 32. Поскольку размерность входных данных PrINN больше, его производительность немного ниже, чем у обычного сегментатора. Однако время выполнения эпохи при этом вырастает всего лишь на 2 секунды. Таким образом, вероятностно-информированные нейросетевые архитектуры демонстрируют схожую вычислительную эффективность с неинформированными аналогами, но при этом сегментируют изображения со значительно более высокой точностью.

Глава 3. Многомасштабное нейросетевое квадродерево для сегментации изображений в условиях сильного дисбаланса разделяемых классов

Глава посвящена разработке информированной НС модели для более точной обработки пространственных связей между элементами изображений в условиях сильно несбалансированных данных. В качестве примера такого рода задачи было решено рассмотреть сегментацию снимков высокого разрешения, содержащих разномасштабные малые объекты.

Глава развивает подход информирования моделью случайного поля Маркова в виде квадродрева, предложенный в главе 2. Свойство эргодичности, доказанное для Марковского поля в виде квадродерева, позволяет обобщать и переносить закономерности, выявленные в разных разрешениях снимка. Согласно теореме 6, информирование моделью квадродерева может быть реализовано на уровне архитектуры с помощью графово-сверточных слоев. Матрица переходных вероятностей поля Маркова в таком случае может рассматриваться как матрица смежности. Такая реализация является новой, потому что существующие подходы использования квадродерева в задачах обработки изображений обычно используют менее явные сверточные представления пространственных связей [204] вместо прямого применения графовых структур. Кроме того, они используются только для специальных задач – оптимизации вычислений в свертках [205], токенизации изображения [206] или реализации многомасштабного внимания [207].

В главе также исследуются аналитические свойства предлагаемой графовой архитектуры, а именно динамика ее обучения. Подробно рассматриваются результаты применения подхода для сегментации несбалансированных наборов аэрокосмических снимков, полученных с помощью спутников и БПЛА, в особенности для обнаружения несбалансированных классов малоразмерных объектов.

3.1 Постановка задачи

Пусть $\mathbb{X} = \{X^k\}_{k=\overline{1,N_X}}$ – набор сегментируемых на K классов изображений, несбалансированный по числу элементов в классе:($N_{(i)} \ll N_{(j)}$, $i = \overline{1,n}$, $j = \overline{n,K}$, $n < K$, где $N_{(i)}$ – число элементов в классе i, а N_X – общее количество элементов набора.

Пусть $X^k_{cat} = (F(X^k)_{inner}, F(X^k), X^k)$, где $F(X^k) \in \mathbb{R}^{H_X \times W_X \times K}$ – результат сегментации снимка базовой нейросетью $F(\cdot)$, а $F(X^k)_{inner}$ – признаки X^k из внутренних слоев $F(\cdot)$, $N_{ch} = ch + K + 3$, ch – число каналов $F(X^k)_{inner}$, а H_X и W_X – высота и ширина обрабатываемых изображений.

Для повышения вероятности правильной сегментации элементов несбалансированных классов ставится задача разработать НС модель $G(\cdot)$:

$$G : \mathbb{R}^{H_X \times W_X \times N_{ch}} \to \mathbb{R}^{H_X \times W_X \times K},$$

которая обрабатывает локальные пространственные связи между внутренними признаками $F(\cdot)$ X^k_{cat} с более высокой точностью за счет информирования вероятностной моделью (Y^k – метки классов пикселей X^k):

$$\mathbb{P}\left(G\left(X^k_{cat}\right) = Y^k\right) > \mathbb{P}\left(F\left(X^k\right) = Y^k\right).$$

Схема НС модели, рассматриваемой в решаемой задаче, представлена на рисунке 3.1.

3.2 Аналитические исследования процесса обучения графовых архитектур, информированных квадродеревом

Графово-сверточные архитектуры широко применяются для обработки данных химических реакций, прогнозирования белков, предсказания социальных реакций [208], а также обработки изображений [209] – сегментации [210] или обнаружения объектов [211]. В этом случае архитектура обычно состоит из кодировщика $F(\cdot)$, формирующего вектор признаков снимка, и графового блока

$$G(\cdot)$$

Рисунок 3.1 — Концепт информированной НС модели для повышения точности распознавания несбалансированных классов

.

Для информирования сети $G(\cdot)$ предлагается использовать модель случайного поля Маркова в виде квадродерева из h слоев, описанного ранее в разделе 2.3. В таком случае, согласно теореме 6, $G(\cdot)$ должна быть информирована на уровне архитектуры и представлять собой графово-сверточную нейронную сеть.

3.2.1 Оценки скорости убывания функции потерь линейных информированных графовых архитектур

Пусть $X \in \mathbb{R}^{H_X \times W_X}$, $X \in \mathbb{X}$ – обрабатываемое изображение. Обозначим множество пикселей X как $X_{i,j}, i = \overline{1, W_X}$, $j = \overline{1, H_X}$, а его вектор пикселей $\mathbf{x} \in \mathbb{R}^{W_X \cdot H_X \times 1}$, элементы которого формируются с помощью правила $x_{i \cdot H_X + j} = X_{i,j}$. Пусть для обработки графовой архитектурой изображение разделяется на одинаковые подобласти, называемые суперпикселями [212], размера $M \times M$, $M \in \mathbb{N}$, $M < min(W_X, H_X)$. Тогда $\mathbf{x}^{(sp)} \in \mathbb{R}^{\frac{W_X \cdot H_X}{M^2} \times M^2}$ – вектор суперпикселей размера M, сформированный из X.

Обозначим $\mathbf{x}_Q$ вектор, содержащий все элементы квадродерева из h слоев. Предположим, что слой S_0 формируется из вектора признаков $\mathbf{x}$, а для построения слоев $S_1, \ldots, S_{h-1}$ используется пулинг по среднему значению, то есть $\mathbf{x}_{\mathbf{Q}}^{(\mathbf{sp})} = \left(avg(\mathbf{x}^{(\mathbf{sp})}), \mathbf{x}^{(\mathbf{sp})}\right) \in \mathbb{R}^{N \times 1}$, где N – общее количество суперпикселей, и $avg(\mathbf{x}^{(\mathbf{sp})}) = \left(avg_h(\mathbf{x}^{(\mathbf{sp})}), \ldots, avg_2(\mathbf{x}^{(\mathbf{sp})})\right)$, и $avg_p(\mathbf{x}) = avg\_pool(\mathbf{x}, 2^p \times 2^p)$, и величина $p = \overline{1, h-1}$ – размер области пулинга. Обозначим $\mathbf{x}_Q^{(sp)}$ вектор признаков квадродерева, разделенный на суперпиксели размера $M \times M$.

Рассмотрим вначале однослойную линейную графовую сеть $G_Q(\cdot)$:

$$G_Q(\mathbf{x}_Q^{(sp)}) = A_Q \cdot \mathbf{x}_Q^{(sp)} \cdot B, \tag{3.1}$$

где $A_Q \in \mathbb{R}^{\frac{N^2}{M^2} \times \frac{N^2}{M^2}}$ – матрица смежности квадродерева, содержащего h слоев, а $B \in M^2 \times M^2$ – матрица линейного преобразования. Покажем, что для $G_Q(\cdot)$ функция потерь убывает быстрее, чем у сопоставимой сети, выполняющей свертку по графу двумерной пиксельной решетки размера $\frac{H_X \cdot W_X}{M^2} \times \frac{H_X \cdot W_X}{M^2}$.

Теорема 8. *Пусть $G(\cdot)$ – однослойная линейная графовая сеть (см. формулу 3.1) с матрицей смежности $A \in R^{\frac{H_X \cdot W_X}{M^2} \times \frac{H_X \cdot W_X}{M^2}}$, соответствующей двумерной пиксельной решетке. Тогда:*

$$\triangle_t \left(L_t(G(\mathbf{x}^{(sp)}), Y)\right) < \triangle_t \left(L_t(G_Q(\mathbf{x}_Q^{(sp)}), Y)\right), \tag{3.2}$$

где $L_t(\cdot, y)$ – произвольная дифференцируемая функция потерь, Y – истинные метки классов, t – номер эпохи обучения, а $\triangle_t$ – изменение величины за одну эпоху обучения (разностная производная).

Доказательство. В статье [213] показано, что для графовой сети со скрытыми связями $G_{skip}(\cdot)$ вида (w_l – значения весов, а H – число последовательных слоев графовой свертки):

$$G_{skip}(\mathbf{x}) = \sum_{l=0}^{H} w_l \cdot X_{(l)}, \tag{3.3}$$

где $X_{(l)} = A \cdot B_{(l)} \cdot X_{(l-1)}$, $X_{(0)} = \mathbf{x}^{(sp)}$, справедливо неравенство

$$\triangle_t \left(L_t(G(\mathbf{x}), \mathbf{y})\right) < \triangle_t \left(L_t(G_{skip}(\mathbf{x}), \mathbf{y})\right), \tag{3.4}$$

описывающее изменения функции потерь за одну эпоху обучения. Для доказательства того, что функция потерь $G_Q(\cdot)$ убывает быстрее, чем у сети $G(\cdot)$, достаточно показать, что $G_Q(\cdot)$ является сетью со скрытыми связями по отношению к сети $G(\cdot)$.

Для доказательства теоремы вначале сделаем ряд предварительных замечаний. Во-первых, свертка изображения X с ядром $w \in \mathbb{R}^{H_w \times H_w}$ может быть представлена как умножение на разреженную матрицу Тёплица [214] W:

$$X * w = W \cdot \mathbf{x}, \tag{3.5}$$

причем на H_w^2 нисходящих диагоналях которой расположены веса w. Тогда пулинг по среднему значению, который по определению эквивалентен двумерной свертке с фиксированным ядром, может быть представлен с помощью линейного преобразования вектора $\mathbf{x}$:

$$\mathbf{x}_Q = U \cdot \mathbf{x}, \tag{3.6}$$

где $U = (U_{2^{h-1}}, \ldots, U_2, I)^T$ – блочная матрица (блоки расположены вдоль диагонали, а остальные элементы заполнены нулями), I – единичная матрица, а U_p – матрица пулинга по среднему значению с полем p, $p = \overline{1,h-1}$. Каждый элемент вектора-результата умножения на матрицу U_p получен как среднее элементов поля $p \times p$ из изображения X. Если X представлено как вектор, то поле $p \times p$ выбирается из $\mathbf{x}$ с помощью умножения на вектор-столбец, в котором расположено p последовательностей из единиц длины p, разделенных $\Theta - 1$ нулями ($\Theta = W_X H_X$).

Во-вторых, обозначим $\mathbf{x}_{flat}$ вектор $\mathbf{x}^{(sp)}$ приведенный к размерности $\Theta \times 1$ с сохранением суперпиксельной структуры. Разница между изображением X и векторами $\mathbf{x}_{flat}$, $\mathbf{x}$ и $\mathbf{x}^{(sp)}$ представлена на рисунке 3.2.

Рисунок 3.2 показывает, что каждые последовательные M^2 элементов $\mathbf{x}_{flat}$ принадлежат одному и тому же суперпикселю. Непосредственным умножением

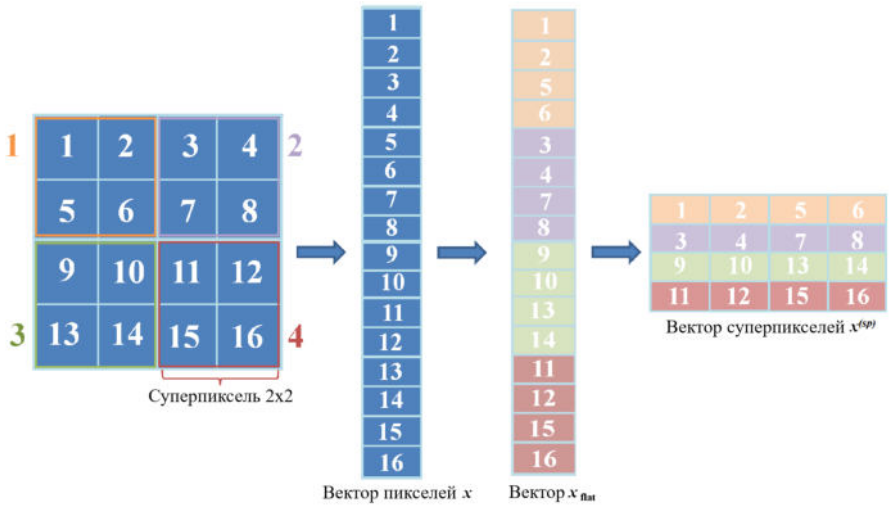

Рисунок 3.2 — Пример формирования векторов $\mathbf{x}_{flat}$, $\mathbf{x}$ и $\mathbf{x}^{(sp)}$ из изображения X $(N = 4,\ M = 2)$

можно показать, что $\mathbf{x}_{flat} = D \cdot \mathbf{x}$, где $D = (D_1, \ldots, D_J)^T$, где $J = \frac{W_X H_X}{M^2}$ и $D_i \in \mathbb{R}^{M^2 \times \Theta}$ имеет следующий вид:

$$D_i = \begin{pmatrix} 0_{M \times M \cdot i},\ I_{M \times M},\ 0_{M \times (\Theta - M \cdot (i+1))} \\ 0_{M \times (H_X + M \cdot i)},\ I_{M \times M}, 0_{M \times (\Theta - M \cdot (i+1) - H_X)} \\ \ldots \\ 0_{M \times (H_X + i) \cdot M},\ I_{M \times M}, 0_{M \times \Theta - (H_X + i + 1) \cdot M} \end{pmatrix}, \tag{3.7}$$

где D_i – блочно-диагональная матрица, включающая M единичных матриц $I_{M \times M}$, сдвинутых друг относительно друга на $H_X \cdot j + i \cdot M$ (здесь j – номер единичной матрицы в D_i). j-я единичная матрица выделяет j-ю строк из i-го суперпикселя, сохраняя порядок следования элементов, а сдвиг до следующей единичной матрицы в D_i равняется сдвигу до $j + 1$ строке суперпикселя в векторе $\mathbf{x}$. Результатом умножения $\mathbf{x}$ на матрицу D_i является вектор длины M^2, состоящий из пикселей i-го суперпиксель, переставленных между собой.

В-третьих, вектор суперпикселей $\mathbf{x}^{(sp)}$ может быть представлен как линейное преобразование вектора $\mathbf{x}_{flat}$ и, соответственно, $\mathbf{x}$:

$$\mathbf{x}^{(sp)} = \sum_{i=1}^{J} \sum_{j=1}^{M^2} (e_i^{(1)})^T \cdot e^{(2)}_{(i-1) \cdot M^2 + j} \cdot \mathbf{x}_{flat} \cdot e_j^{(3)}, \tag{3.8}$$

где $e_i^{(k)}$, $k = \overline{1,3}$ базисные вектора, и $e_i^{(1)} \in \mathbb{R}^{1 \times J}$, $e_i^{(2)} \in \mathbb{R}^{1 \times \Theta}$ и $e_i^{(3)} \in \mathbb{R}^{1 \times M^2}$. Кроме того, для вектора $\mathbf{x}^{(sp)}$ и матриц $A = \{a_{kl}\} \in \mathbb{R}^{J \times J}, B = \{b_{kl}\} \in \mathbb{R}^{M^2 \times M^2}$

непосредственным умножением можно показать справедливость следующих выражений:

$$A \cdot \mathbf{x}^{(sp)} \cdot B = \sum_{i=1}^{J} \sum_{j=1}^{M^2} (e_i^{(1)})^T \cdot e^{(2)}_{(i-1)\cdot M^2+j} \cdot A^{flat} B^{flat} \cdot D \cdot \mathbf{x} \cdot e_j^{(3)}, \tag{3.9}$$

где $A^{flat} = \{A_k^{flat}\}, B^{flat} = \{B_k^{flat}\}, B_k^{flat} \in \mathbb{R}^{\frac{N^2}{M^2} \times N^2}, A_k^{flat} \in \mathbb{R}^{\frac{N^2}{M^2} \times N^2}$, $k = \overline{1,J}$, при этом:

$$A_k^{flat} = \begin{pmatrix} a_{k1}, \underbrace{0..0}_{M^2-1}, a_{k2}, .., a_{kJ}, 0..0 \\ 0, a_{k1}, \underbrace{0..0}_{M^2-1}, a_{k2}, .., a_{kJ}, 0..0 \\ \dots \\ \underbrace{0..0}_{M^2-1}, a_{k1}, \underbrace{0..0}_{M^2-1}, a_{k2}, ...a_{kJ} \end{pmatrix}, \tag{3.10}$$

$$B_k^{flat} = \begin{pmatrix} \underbrace{0..0}_{k\cdot M^2}, b_{11}..b_{M1}, b_{M+1,1}..b_{M^2,1}, ..0 \\ \dots \\ \underbrace{0..0}_{k\cdot M^2}, b_{1M}..b_{MM}, b_{M+1,M}..b_{M^2,M}, ..0 \end{pmatrix}. \tag{3.11}$$

Перейдем к доказательству основного утверждения теоремы. Матрица смежности A двумерной решетки описывает пространственные взаимосвязи между элементами в S_0 слое квадродерева. Тогда матрица смежности квадродерева A_Q может быть представлена в виде суммы двух матриц: $A_Q = A_q + \begin{pmatrix} 0 & 0 \\ 0 & A \end{pmatrix}$. Аналогично формуле (3.6) вектор признаков квадродерева $\mathbf{x}_Q^{(sp)} = U \cdot \mathbf{x}^{(sp)}$, и тогда $G_Q(\mathbf{x}_Q^{(sp)})$ имеет вид:

$$G_Q(\mathbf{x}_Q^{(sp)}) = A_Q \cdot \mathbf{x}_Q^{(sp)} \cdot B = A_q \cdot \mathbf{x}_Q^{(sp)} \cdot B + \begin{pmatrix} 0 & 0 \\ 0 & A \end{pmatrix} \cdot \mathbf{x}_Q^{(sp)} \cdot B. \tag{3.12}$$

В этой формуле

$$\begin{pmatrix} 0 & 0 \\ 0 & A \end{pmatrix} \cdot \mathbf{x}_Q^{(sp)} = \begin{pmatrix} 0 & 0 \\ 0 & A \end{pmatrix} \cdot \begin{pmatrix} 0 \\ \mathbf{x}^{(sp)} \end{pmatrix} = \begin{pmatrix} 0 \\ I_{J\times J} \end{pmatrix} \cdot A \cdot \mathbf{x}^{(sp)}. \tag{3.13}$$

Кроме того, $A_q \cdot \mathbf{x}_Q^{(sp)} \cdot B = A_q \cdot U \cdot \mathbf{x}^{(sp)} \cdot B = A_{qU} \cdot \mathbf{x}^{(sp)} \cdot B$. Тогда:

$$G_Q(\mathbf{x}_Q^{(sp)}) = (A_{qU} \cdot \mathbf{x}^{(sp)} + \begin{pmatrix} 0 \\ I_{J\times J} \end{pmatrix} \cdot A \cdot \mathbf{x}^{(sp)}) \cdot B = \tag{3.14}$$

$$= A_{qU} \cdot \mathbf{x}^{(sp)} \cdot B + \begin{pmatrix} 0 \\ I_{J\times J} \end{pmatrix} \cdot G(\mathbf{x}^{(sp)}).$$

В соответствии с формулами (3.3) и (3.9) при $H = 1$, выражение $A_{qU} \cdot \mathbf{x}^{(sp)} \cdot B$ может быть обозначено как $w_0 \cdot \mathbf{x}^{(sp)}$ за счет переноса слагаемых в (3.9). Поэтому $G_Q(\cdot)$ является графово-сверточной сетью со скрытыми связями и матрицей смежности A. □

Представленная выше теорема демонстрирует, что использование квадродеревьев для описания пространственных взаимосвязей между пикселями улучшает процесс обучения графовых нейронных сетей в сравнении с архитектурами, в которых пространственные связи описываются двумерной решеткой.

3.2.2 Оценки скорости убывания функции потерь информированных квадродеревом графовых архитектур с обработкой локальных связей

Рассмотрим обработку вектора признаков $\mathbf{x}^{(sp)}$ в графовой сети со скрытыми связями при $H = 1$ (формула (3.9)). В ветви скрытых связей выполняется линейное преобразование $w_0 \cdot \mathbf{x}^{(sp)}$, которое может быть представлено линейной графовой сверткой с матрицей $V = I_{N\times N}$, такое что $w_0 \cdot \mathbf{x}^{(sp)} = w_0 \cdot V \cdot \mathbf{x}^{(sp)}$.

Матрицы V и A_Q, описывающие структуру квадродерева, различны. Следовательно, при обработке $\mathbf{x}^{(sp)}$ задействовано две марковские модели. Рассмотрим случай, когда $V \neq I_{N\times N}$. Если графы, описываемые матрицами V и A_Q, не содержат общих ребер, то будем говорить, что для обработки $\mathbf{x}^{(sp)}$ используется мультикомпонентное поле Маркова, построенное по графу ζ, состоящему из $\beta = 2$ подграфов. Поскольку граф A_Q описывает глобальные взаимосвязи и $V \neq I_{N\times N}$, естественным будет выбор в качестве V графа локальных взаимосвязей между признаками изображений.

Пусть графом, описываемым матрицей V, является двумерная пиксельная решетка размера $M \times M$, задающая внутрисуперпиксельные связи. Тогда

графовой архитектурой, информированной мультикомпонентным полем Маркова является $G^*_Q(\cdot)$ вида:

$$G^*_Q(\mathbf{x}) = w_1 \cdot \mathbf{x_{gRes}} + w_2 \cdot \mathbf{x_{lRes}} \tag{3.15}$$

где $\mathbf{x_{gRes}}$ – результат обработки глобальных признаков, а $\mathbf{x_{lRes}}$ – локальных (индекс $lRes$ означает, что вектор является результатом обработки локальных признаков, а $gRes$ – глобальных).

Линейный блок обработки глобальных признаков

Рассмотрим вначале случай, когда обработка глобальных признаков реализована линейным образом:

$$\mathbf{x}^{(\mathbf{sp})}_{\mathbf{gRes}} = GCN_{res}(\mathbf{x}^{(\mathbf{sp})}_{\mathbf{Q}}) = \alpha \cdot \mathbf{x}^{(\mathbf{sp})}_{\mathbf{Q}\text{-}*} + A_Q \mathbf{x}^{(\mathbf{sp})}_{\mathbf{Q}\text{-}*} B_Q, \tag{3.16}$$

где B_Q– матрица линейного преобразования глобальных признаков, $\alpha \in [0,1]$, а $\mathbf{x}^{(\mathbf{sp})}_{\mathbf{Q}\text{-}*} = U \cdot \mathbf{x}^{(\mathbf{sp})}_{\mathbf{pool}}$, и вектор $\mathbf{x_{pool}}$, разбитый на суперпиксели размера M? имеет вид:

$$\mathbf{x}^{(\mathbf{sp})}_{\mathbf{pool}} = U_q \cdot \mathbf{x}, \;\; q = \overline{1,2} \tag{3.17}$$

Формула (3.16) позволяет учесть влияние стандартных модификаций графовых слоев, таких как дополнительная скрытая связь (коэффициент α) и дополнительное сжатие входных данных для уменьшения влияния шума (вектор $\mathbf{x_{pool}}$), на обработку глобальных признаков.

Пусть

$$\mathbf{x}^{(\mathbf{sp})}_{\mathbf{lRes}} = \mathbf{x}^{(\mathbf{sp})}_{\mathbf{local}} \cdot V \cdot B_{loc}, \tag{3.18}$$

где $\mathbf{x}^{(\mathbf{sp})}_{\mathbf{local}} = U \cdot \mathbf{x}^{(\mathbf{sp})}$. Слагаемые $\mathbf{x_{gResEx}}$ и $\mathbf{x_{lResEx}}$ из формулы (3.15) формируются из векторов $\mathbf{x}^{(\mathbf{sp})}_{\mathbf{gRes}}$ и $\mathbf{x}^{(\mathbf{sp})}_{\mathbf{lRes}}$ соответственно. Если в формуле (3.17) степень сжатия $q = 1$, то $\mathbf{x_{lResEx}} = \mathbf{x}^{(\mathbf{sp})}_{\mathbf{lRes}}$ и $\mathbf{x_{gResEx}} = \mathbf{x}^{(\mathbf{sp})}_{\mathbf{gRes}}$.

Если $q = 2$, то по определению $\mathbf{x}^{(\mathbf{sp})}_{\mathbf{lRes}}$ и $\mathbf{x}^{(\mathbf{sp})}_{\mathbf{gRes}}$ имеют разную размерность: в векторе $\mathbf{x}^{(\mathbf{sp})}_{\mathbf{gRes}}$ слой S_0 квадродерева высоты h строится по $\mathbf{x}^{(\mathbf{sp})}_{\mathbf{pool}}$ и содержит в четыре раза меньше элементов, чем слой S_0 в $\mathbf{x}^{(\mathbf{sp})}_{\mathbf{lRes}}$. Необходимо привести векторы к единой размерности, при этом объединяя признаки по слоям для

сохранения структуры квадродерева. Для этого каждый вектор должен быть разделен на h подвекторов $\mathbf{x_{gRes}}^{(i)}$ и $\mathbf{x}_{lRes}^{(i)}$, $i = \overline{1,h}$, в соответствии с принадлежностью элементов к S_i. Далее эти подвекторы приводятся к одному масштабу и конкатенируются: $\mathbf{x_{cat}}^{(i)} = \text{concat}(\mathbf{x_{lResEx}}^{(i)}, \mathbf{x_{gResEx}}^{(i)})$, $i = \overline{1,h+1}$, где $\mathbf{x_{lResEx}} = \left(0_{\frac{N^2}{2^h M^2} \times M^2}, \mathbf{x}_{lRes}\right)$ и $\mathbf{x_{gResEx}} = \left(\mathbf{x_{gRes}}, 0_{\frac{N^2}{M^2} \times M^2}\right)$ – дополненные нулями векторы результатов обработки глобальных и локальных признаков в графовом блоке. Дополнение может быть реализовано за счет умножения на матрицы $I_{loc} = \left(0_{\frac{N^2}{2^h M^2} \times \frac{N^2}{M^2}}, I_{\frac{N^2}{M^2} \times \frac{N^2}{M^2}}\right)$ и $I_{glob} = \left(I_{\frac{N^2}{4M^2} \times \frac{N^2}{4M^2}}, 0_{\frac{N^2}{M^2} \times \frac{N^2}{4M^2}}\right)$.

Тогда:

$$G_Q^*(\mathbf{x}) = w_1 \cdot \mathbf{x_{gResEx}} + w_2 \cdot \mathbf{x_{lResEx}} = w_2 \cdot I_{loc}\mathbf{x_{lRes}} + w_1 \cdot I_{glob}\mathbf{x_{gRes}} = \\ = w_1 \cdot I_{glob}(\mathbf{x_Q^{(sp)}} + A_Q \mathbf{x_Q^{(sp)}} B_Q) + w_2 \cdot I_{loc} \cdot \mathbf{x_{local}^{(sp)}} V \cdot B. \quad (3.19)$$

Теорема 9. *Пусть заданы нейронные сети $G_Q^*(\cdot)$ (информированная сеть, формула (3.19)), сверточная $F(\mathbf{x}) = X * w$ и линейная графовая $G(\mathbf{x}) = A \cdot \mathbf{x}^{(sp)} \cdot B_G$, причем $G_Q^*(\cdot)$ и $G(\cdot)$ обрабатывают изображение по суперпикселям размера M. Пусть $L_t(\cdot, \mathbf{y})$ – дифференцируемая функция потерь. Тогда справедливы следующие неравенства:*

$$\triangle_t \left(L_t(G(\mathbf{x}), \mathbf{y})\right) < \triangle_t \left(L_t(G_Q^*(\mathbf{x}), \mathbf{y})\right), \\ \triangle_t \left(L(F(\mathbf{x}), \mathbf{y})\right) < \triangle_t \left(L(G_Q^*(\mathbf{x}), \mathbf{y})\right), \quad (3.20)$$

где $\mathbf{y}$ – вектор истинных значений, а t задает шаг обучения сети, то есть $G_Q^*(\cdot)$ обучается быстрее, чем сопоставимые по размеру графовые и сверточные решения.

Доказательство. Для доказательства того, что функция потерь $G_Q^*(\cdot)$ убывает быстрее в сравнении с сопоставимыми архитектурами $F(\cdot)$ и $G(\cdot)$ каждую эпоху обучения, достаточно показать, что $G_Q^*(\cdot)$ является сетью со скрытыми связями по отношению к сетям $F(\cdot)$ и $G(\cdot)$. Имеем:

$$G_Q^*(\mathbf{x}) = w_2 \cdot I_{loc} \cdot U\mathbf{x}^{(sp)}V \cdot B + w_1 \cdot I_{glob}(\boldsymbol{\alpha} \cdot U\mathbf{x_{pool}^{(sp)}} + A_Q U \mathbf{x_{pool}^{(sp)}} B_Q).$$

Матрица смежности квадродерева A_Q выражается через матрицу A, описывающую связи между элементами нижнего слоя квадродерева: $A_Q = A_q +$

$\begin{pmatrix} 0 & 0 \\ 0 & A \end{pmatrix}$. Получим:

$$G_{pool}(\mathbf{x}) = A_Q U \mathbf{x}^{(\mathbf{sp})}_{\mathbf{pool}} \cdot B_Q = A_q U \mathbf{x}^{(\mathbf{sp})}_{\mathbf{pool}} \cdot B_Q + \begin{pmatrix} 0 & 0 \\ 0 & A \end{pmatrix} \cdot U \mathbf{x}^{(\mathbf{sp})}_{\mathbf{pool}} \cdot B_Q =$$

$$= A_q U \mathbf{x}^{(\mathbf{sp})}_{\mathbf{pool}} \cdot B_Q +$$

$$+ \begin{pmatrix} 0 & 0 \\ 0 & A \end{pmatrix} \cdot \begin{pmatrix} 0 \\ \mathbf{x}^{(\mathbf{sp})}_{\mathbf{pool}} \end{pmatrix} B_Q = (A_q U \mathbf{x}^{(\mathbf{sp})}_{\mathbf{pool}} + \begin{pmatrix} 0 \\ I_{\frac{J}{4} \times \frac{J}{4}} \end{pmatrix} A \cdot \mathbf{x}^{(\mathbf{sp})}_{\mathbf{pool}}) B_Q =$$

$$= A_q U \cdot \mathbf{x}^{(\mathbf{sp})}_{\mathbf{pool}} \cdot B_Q + \begin{pmatrix} 0 \\ I_{\frac{J}{4} \times \frac{J}{4}} \end{pmatrix} \cdot G(\mathbf{x}^{(\mathbf{sp})}_{\mathbf{pool}}).$$

Согласно формуле (3.3) при $H = 1$, $G_{pool}(\mathbf{x})$ – графово-сверточная нейронная сеть со скрытыми связями по отношению к $G(\cdot)$. Далее имеем:

$$G^*_Q(\mathbf{x}) = w_2 \cdot I_{loc} \cdot U \sum_{i=1}^{N} \sum_{j=1}^{M^2} e^{(1)}_i \cdot e^{(2)}_{i \cdot M^2 + j} \cdot D \cdot \mathbf{x} \cdot e^{(3)}_j \cdot V \cdot B +$$

$$+ w_1 \cdot I_{glob} (\alpha \cdot U \sum_{i=1}^{\frac{N}{4}} \sum_{j=1}^{M^2} e^{(1)}_i \cdot e^{(2)}_{i \cdot M^2 + j} \cdot D \cdot \mathbf{x} \cdot e^{(3)}_j + w_1 \cdot I_{glob} G_{pool}(\mathbf{x}^{(\mathbf{sp})}_{\mathbf{pool}})) =$$

$$= \sum_{i=1}^{N+2} \sum_{j=1}^{M^2} e^{(1)}_i \cdot e^{(2)}_{i \cdot M^2 + j} \underbrace{(w_2^{flat} I_{loc}^{flat} U^{flat} \cdot V^{flat} B^{flat} \cdot D + w_1^{flat} I_{glob}^{flat} D)}_{w_4^{flat}} \cdot \mathbf{x} \cdot e^{(3)}_j +$$

$$+ w_1 \cdot I_{glob} G_{pool}(\mathbf{x}^{(\mathbf{sp})}_{\mathbf{pool}}) =$$

$$= \sum_{i=1}^{N+2} \sum_{j=1}^{M^2} e^{(1)}_i \cdot e^{(2)}_{i \cdot M^2 + j} (w_4^{flat} \cdot \mathbf{x} + \underbrace{w_1^{flat} \cdot I_{glob}^{flat}}_{w_5^{flat}} A_Q^{flat} \underbrace{U^{flat} B_Q^{Flat} D U_2}_{B_{Q\text{-}ex}^{flat}} \mathbf{x}) \cdot e^{(3)}_j =$$

$$= \sum_{i=1}^{N+2} \sum_{j=1}^{M^2} e^{(1)}_i \cdot e^{(2)}_{i \cdot M^2 + j} (w_4^{flat} \cdot \mathbf{x} + w_5^{flat} \underbrace{A_Q^{flat} B_{Q\text{-}ex}^{flat} \mathbf{x}}_{G_{pool}(\mathbf{x})} \cdot e^{(3)}_j.$$

Из формулы (3.3) вытекает, что $G^*_Q(\mathbf{x})$ – сеть со скрытыми связями по отношению к $G_{pool}(\mathbf{x}^{(\mathbf{sp})}_{\mathbf{pool}})$ и, соответственно, по отношению к $G(\mathbf{x})$. При этом $G^*_Q(\mathbf{x}) = (w_4^{flat} + w_5^{flat} A_Q^{flat} B_{Q\text{-}ex}^{flat})\mathbf{x} - I_{loc} U \cdot D \cdot W \cdot \mathbf{x} + I_{loc} U \cdot D \cdot W \cdot \mathbf{x} = w_6^{flat} \mathbf{x} + w_7^{flat} F(\mathbf{x})$. Следовательно, $G^*_Q(\mathbf{x})$ – графово-сверточная сеть со скрытыми связями ($H = 1$) относительно $F(\mathbf{x})$.

Поскольку $G^*_Q(\mathbf{x})$ является сетью со скрытыми связями относительно $F(\mathbf{x})$ и $G(\mathbf{x})$, то справедливы оба неравенства (9). □

Из теоремы 9 вытекает, что использование модели мультикомпонентного случайного поля Маркова, одной из компонент которого является квадродерево, для описания глобальных и локальных взаимосвязей между элементами изображения, повышает скорость обучения линейной графовой сети в сравнении с сопоставимыми линейными графовыми и сверточными архитектурами.

Нелинейный блок обработки глобальных признаков

Рассмотрим теперь случай, когда обработка глобальных признаков реализована с помощью графового слоя с самовниманием (обработка локальных признаков, как и в линейном случае, проводится по формуле (3.18)):

$$\mathbf{x}^{(\mathbf{sp})}_{\mathbf{gRes}} = G_{Att}(\mathbf{x}^{(\mathbf{sp})}_{\mathbf{Q}}) = w_1 \cdot A_Q \odot sfm(\mathbf{x}^{(\mathbf{sp})}_{\mathbf{Q}\text{-}*} \cdot Q \cdot (\mathbf{x}^{(\mathbf{sp})}_{\mathbf{Q}\text{-}*})^T)\mathbf{x}^{(\mathbf{sp})}_{\mathbf{Q}\text{-}*} B_Q, \tag{3.21}$$

где $sfm(\cdot) = softmax(\cdot)$, а Q – матрица, соответствующая $\frac{W_q \cdot W_k^T}{\sqrt{d_k}}$ из формулы (2), представленной ранее во введении.

Теорема 10. *Пусть заданы информированная нейронная сеть* $G^*_Q(\mathbf{x}) = \mathbf{x}_{\mathbf{gRes}} + w_2 \cdot \mathbf{x}_{\mathbf{lRes}}$, *глобальные признаки которой* $\mathbf{x}_{\mathbf{gRes}} = G_{Att}(\mathbf{x}^{(\mathbf{sp})}_{\mathbf{Q}})$, *а также графовая сеть с вниманием* $G_{Att}(\cdot)$ *(формула (3.21)), причем обе сети обрабатывают изображение по суперпикселям размера* M. *Пусть* $L_t(\cdot, \mathbf{y})$ – *дифференцируемая функция потерь. Тогда*

$$\triangle_t \left(L_t(G_{Att}(\mathbf{x}), \mathbf{y})\right) < \triangle_t \left(L_t(G^*_Q(\mathbf{x}), \mathbf{y})\right),$$

то есть $G^*_Q(\cdot)$ *обучается быстрее, чем с* $G_{Att}(\cdot)$.

Доказательство. Согласно определению $G^*_Q(\mathbf{x}) = G_{Att}(\mathbf{x}^{(\mathbf{sp})}_{\mathbf{Q}}) + w_2 \cdot \mathbf{x}^{(\mathbf{sp})}_{\mathbf{local}} V \cdot B$. Распишем $\triangle_t \left(L_t(\hat{y}, \mathbf{y})\right)$, где $\hat{y}$ – предсказанное значение:

$$\triangle_t \left(L_t(\hat{y}, \mathbf{y})\right) = \triangle_t \left(\hat{y}\right) \triangle_{\hat{y}} \left(L\right). \tag{3.22}$$

Здесь и далее при вычислении производных будем обозначать $L_t(\cdot)$ как $L(\cdot)$. Рассмотрим первый множитель и подставим $G^*_Q(\mathbf{x})$ вместо $\hat{y}$:

$$\triangle_t \left(G^*_Q(\mathbf{x})\right) = \triangle_t \left(G_{Att}(\mathbf{x}^{(\mathbf{sp})}_{\mathbf{Q}})\right) + \triangle_t \left(w_2 \cdot \mathbf{x}^{(\mathbf{sp})}_{\mathbf{local}} V \cdot B\right). \tag{3.23}$$

Следуя подходу, представленному в работе [213], оценим второе слагаемое. Заметим, что согласно формуле (3.9):

$$w_2 \cdot \mathbf{x}_{\mathbf{local}}^{(\mathbf{sp})} V \cdot B = \sum_{i=1}^{J} \sum_{j=1}^{M^2} (e_i^{(1)})^T \cdot e_{(i-1)\cdot M^2+j}^{(2)} \cdot w_2^{flat} U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x} \cdot e_j^{(3)}. \quad (3.24)$$

Умножение на базисные векторы имеет целью только корректирование размерности и не приводит к изменению значений параметров матрицы $w_2^{flat} U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x}$. Поэтому:

$$\triangle_t \left(w_2 \cdot \mathbf{x}_{\mathbf{local}}^{(\mathbf{sp})} V \cdot B \right) = \sum_{i=1}^{J} \sum_{j=1}^{M^2} (e_i^{(1)})^T \cdot e_{(i-1)\cdot M^2+j}^{(2)} \cdot \triangle_t \left(w_2^{flat} U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x} \right) \cdot e_j^{(3)}.$$

Обозначим $\mathbf{W} = w_2^{flat} U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x}$. Производная $\triangle_t$ является разностной и для дальнейшего доказательства теоремы достаточно рассмотреть ее линейную аппроксимацию ($\varepsilon \to 0$):

$$\triangle_t (\mathbf{W}) = \frac{\mathbf{W}' - \mathbf{W}}{\varepsilon}, \quad (3.25)$$

где $\mathbf{W}$ – это значение параметров на t эпохе обучения, а $\mathbf{W}'$ – на $t+1$-й. При этом $\mathbf{W}$ представляет собой произведение нескольких матриц весов нейронной сети. Поэтому $\mathbf{W}'$ может быть вычислена по правилу дифференцирования произведения функций:

$$\mathbf{W}' = (w_2^{flat})' U^{flat} V^{flat} \cdot (B^{flat})' \cdot D \cdot \mathbf{x}, \quad (3.26)$$

Матрицы U^{flat}, V^{flat}, D и вектор $\mathbf{x}$ на каждой итерации имеют одинаковые значения. А на $t+1$-й эпохе $(w_2^{flat})'$ и $(B^{flat})'$ могут быть выражены через w_2^{flat} и B^{flat} и значение градиента функции потерь $L(\cdot)$ по этим параметрам (в формуле представлено его разложение по правилу вычисления частных производных):

$$(w_2^{flat})' = w_2^{flat} - \varepsilon \, \triangle_{\hat{y}} \, L \, \triangle_{\hat{w}_2^{flat}} \, \hat{y}, \quad (B^{flat})' = B^{flat} - \varepsilon \, \triangle_{\hat{y}} \, L \, \triangle_{\hat{B}^{flat}} \, \hat{y}. \quad (3.27)$$

Вычислим $\triangle_{\hat{w}_2^{flat}} \, \hat{y}$ и $\triangle_{\hat{B}^{flat}} \, \hat{y}$. Из формул (3.25) и (3.23) для этого нужно рассмотреть выражения $\triangle_{\hat{w}_2^{flat}} \left(w_2^{flat} U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x} \right)$ и $\triangle_{\hat{B}^{flat}} \left(w_2^{flat} U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x} \right)$. Продифференцируем их по правилам вычисления тензорной производной (см., например, в [213]):

$$\triangle_{\hat{w}_2^{flat}} \left(w_2^{flat} U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x}\right) =$$
$$= \triangle_{\hat{w}_2^{flat}} \left(I_{N \cdot M \times N \cdot M} \otimes U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x}\right) =$$
$$= \left(U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x}\right)^T \otimes I_{N \cdot M \times N \cdot M},$$
$$\frac{\partial (w_2^{flat} U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x})}{\partial B^{flat}} =$$
$$= \triangle_{w_2^{flat} U^{flat} V^{flat} \cdot B^{flat}} \left(w_2^{flat} U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x}\right) \triangle_{\hat{B}^{flat}} \left(w_2^{flat} U^{flat} V^{flat} \cdot B^{flat}\right) =$$
$$= (D \cdot \mathbf{x})^T \odot I_{H_X \times W_X} w_2^{flat} U^{flat} V^{flat} \odot I_{H_X \times W_X}.$$

Подставим полученные выражения в (3.26) и преобразуем получающиеся слагаемые с учетом того, что норма $||\cdot||_2$ может быть выражена через скалярное произведение векторов:

$$\mathbf{W}' = (w_2^{flat} - \varepsilon \triangle_{\hat{y}} L (U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x})^T \otimes I_{N \cdot M \times N \cdot M}) U^{flat} V^{flat} \cdot (B^{flat} -$$
$$- \varepsilon \triangle_{\hat{y}} L (D \cdot \mathbf{x})^T \odot (I_{H_X \times W_X} w_2^{flat} U^{flat} V^{flat} \odot (I_{H_X \times W_X}) \cdot D \cdot \mathbf{x} =$$
$$= w_2^{flat} U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x} -$$
$$- \varepsilon \triangle_{\hat{y}} L ((U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x})^T \otimes I_{N \cdot M \times N \cdot M} U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x} +$$
$$+ w_2^{flat}) U^{flat} V^{flat} (D \cdot \mathbf{x})^T \odot I_{H_X \times W_X} w_2^{flat} U^{flat} V^{flat} \odot I_{H_X \times W_X} \cdot D \cdot \mathbf{x}) + O(\varepsilon^2) =$$
$$= w_2^{flat} U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x} - \varepsilon \triangle_{\hat{y}} L (||U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x}||_2 +$$
$$+ (D \cdot \mathbf{x})^T \odot I_{H_X \times W_X} (w_2^{flat}) U^{flat} V^{flat})^T w_2^{flat} U^{flat} V^{flat} \odot I_{H_X \times W_X} \cdot D \cdot \mathbf{x}) + O(\varepsilon^2) =$$
$$= w_2^{flat} U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x} - \varepsilon \triangle_{\hat{y}} L (||U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x}||_2 +$$
$$+ ||w_2^{flat} U^{flat} V^{flat} \cdot D \cdot \mathbf{x}||_2) + O(\varepsilon^2).$$

Подставим получившееся выражение в формулу (3.25) и устремим ε к нулю. Отсюда:

$$\triangle_t (\mathbf{W}) = - \triangle_{\hat{y}} L (||U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x}||_2 + ||w_2^{flat} U^{flat} V^{flat} \cdot D \cdot \mathbf{x}||_2)$$

Выражение в скобках есть сумма двух норм и потому строго положительно. Сравним теперь $\triangle_t \left(L(G_Q^*(\mathbf{x}), \mathbf{y})\right)$ и $\triangle_t \left(L(G_{Att}(\mathbf{x}_{\mathbf{Q}}^{(\mathbf{sp})}), \mathbf{y})\right)$. Вычтем одну производную из другой:

$$\triangle_{\hat{y}} L \triangle_t (G_Q^*(\mathbf{x})) - \triangle_{\hat{y}} L \triangle_t (G_{Att}(\mathbf{x}_{\mathbf{Q}}^{(\mathbf{sp})})) = \triangle_{\hat{y}} L \triangle_t (w_2 \cdot \mathbf{x}_{\mathbf{local}}^{(\mathbf{sp})} V \cdot B) =$$
$$= -(|| \triangle_{\hat{y}} L U^{flat} V^{flat} \cdot B^{flat} \cdot D \cdot \mathbf{x}||_2 + || \triangle_{\hat{y}} L w_2^{flat} U^{flat} V^{flat} \cdot D \cdot \mathbf{x}||_2) < 0$$

Таким образом, дифференцируемая функция потерь от $G_Q^*(\mathbf{x})$ обучается быстрее, чем $G_{Att}(\mathbf{x}_{\mathbf{Q}}^{(\mathbf{sp})})$. □

Теорема 10 означает, что использование дополнительной ветви для обработки локальных признаков позволяет повысить скорость обучения $G_Q^*(\cdot)$ по сравнению с сопоставимой графовой сетью с вниманием.

Использование механизма внимания в $G_{Att}(\cdot)$ эквивалентно повышению степени интерполяционного полинома для восстановления данных. Например, для линейной $G(\cdot)$ степень полинома равняется 1, а $G_{Att}(\cdot)$ напротив, может быть представлена в виде полинома любой степени выше или равной первой. Функция $sfm(\mathbf{z}Q\mathbf{z})$, входящая в $G_{Att}(\cdot)$, может быть разложена в бесконечный ряд Тейлора. Действительно, первая производная $sfm'(\mathbf{z}) = sfm(\mathbf{z})(I - sfm(\mathbf{z}))$ включает в себя функцию $sfm(\cdot)$. Производная порядка n также будет включать в себя функцию $sfm(\cdot)$, а значит на ее основе может быть вычислена производная порядка $n+1$, не равная нулю. Если $sfm(\mathbf{z})$ на основе разложения по Тейлору представляется полиномом степени n, то выражение $sfm(\mathbf{z})\mathbf{z}$ – полиномом степени $n + 1 \geqslant 1$.

Представленные выше утверждения объясняют наблюдаемое в экспериментах повышение точности восстановления данных при использовании графовых сетей с вниманием в сравнении с линейными, представленное в дальнейших разделах. Однако не представляется возможным сделать однозначные общие выводы о скорости обучения архитектур $G_{Att}(\cdot)$ и $G(\cdot)$ – сети с вниманием и линейной графовой сети, аналогичные представленным в теоремах 10 и 9. $G_{Att}(\cdot)$ и $G(\cdot)$ представляются полиномами разной степени. Аналогично случаю обычных степенных функций $f(z) = z^l, l = \overline{1,\infty}$, величина производной и, соответственно, превосходство в скорости обучения одной функции над другой не постоянно и зависит от значений обучаемых параметров и аргумента.

3.3 Архитектура FN-QiGSAN

На основе теорем, доказанных в разделе 3.2, была разработана новая информированная ансамблевая графовая сеть Fused-Nested quadtree informed Graph Self-Attention Network (FN-QiGSAN), представленная на рисунке 3.3. В FN-QiGSAN входят кодировщик $F(\cdot)$ и графовая сеть $G(\cdot)$, обрабатывающая внутренние признаки изображения, формируемые кодировщиком. $G(\cdot)$ информирована на уровне архитектуры моделью мультикомпонентного поля

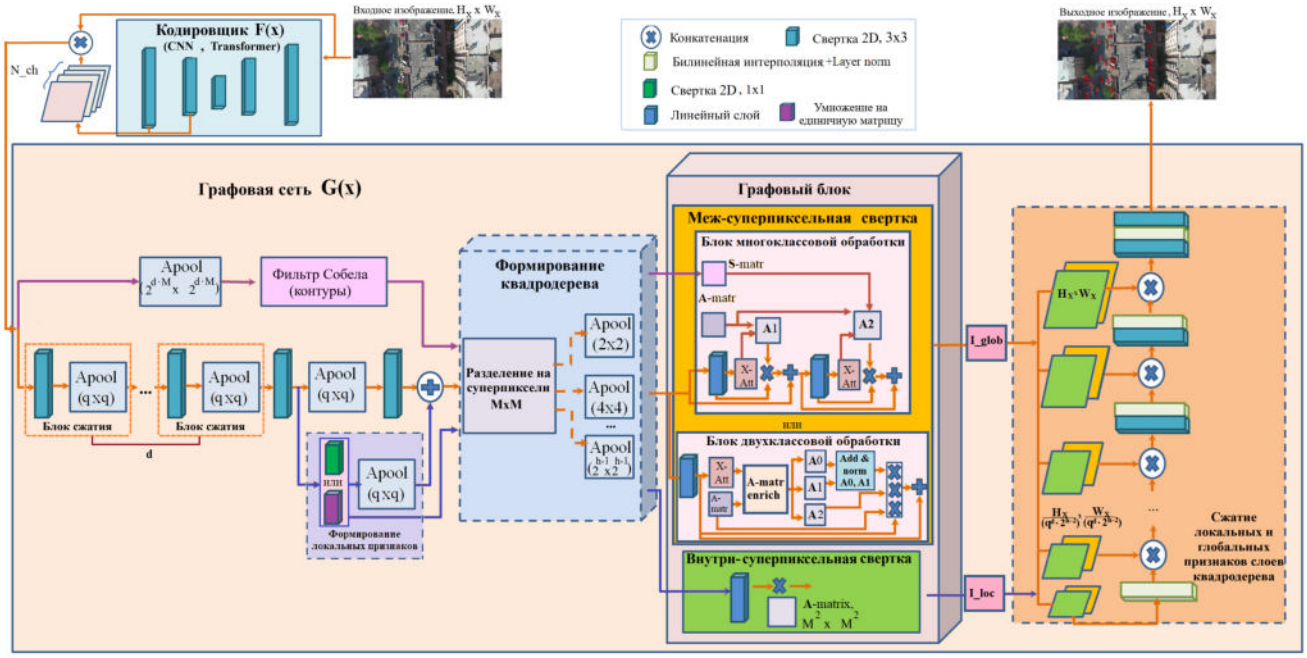

Рисунок 3.3 — Архитектура FN-QiGSAN

Маркова, граф ζ которого состоит из $\beta = 2$ несвязанных ребрами подграфов, моделирующих локальные и глобальные взаимосвязи. Граф локальных взаимосвязей – двумерная пиксельная решетка: каждый пиксель связан с 8 окружающими его ближайшими соседями по горизонтали, вертикали и двум диагоналям. Граф глобальных взаимосвязей имеет форму квадродерева.

В FN-QiGSAN информирование моделью случайного поля Маркова реализовано в блоке графовых сверток, а также в сопутствующих ему блоках формирования структуры квадродерева и объединения глобальных и локальных признаков (выделены пунктирными линиями на рисунке 3.3). Матрицы смежности локальных и глобальных признаков, используемые в «Графовом блоке», соответствуют матрицам переходных вероятностей поля Маркова. Переходы между элементами поля изначально считаются равновероятными: настройка значений элементов матриц реализуется в процессе обучения сети. При этом положение ненулевых элементов остается фиксированным в соответствии со структурой квадродерева, что и реализует информирование сети на уровне архитектуры.

Конфигурация сети FN-QiGSAN определяется размером обрабатываемых изображений, а также решаемой задачей. Эти параметры определяют степень сжатия глобальных признаков, а также выбор графового блока для их обработки (см. разделы 3.3.1 и 3.3.2).

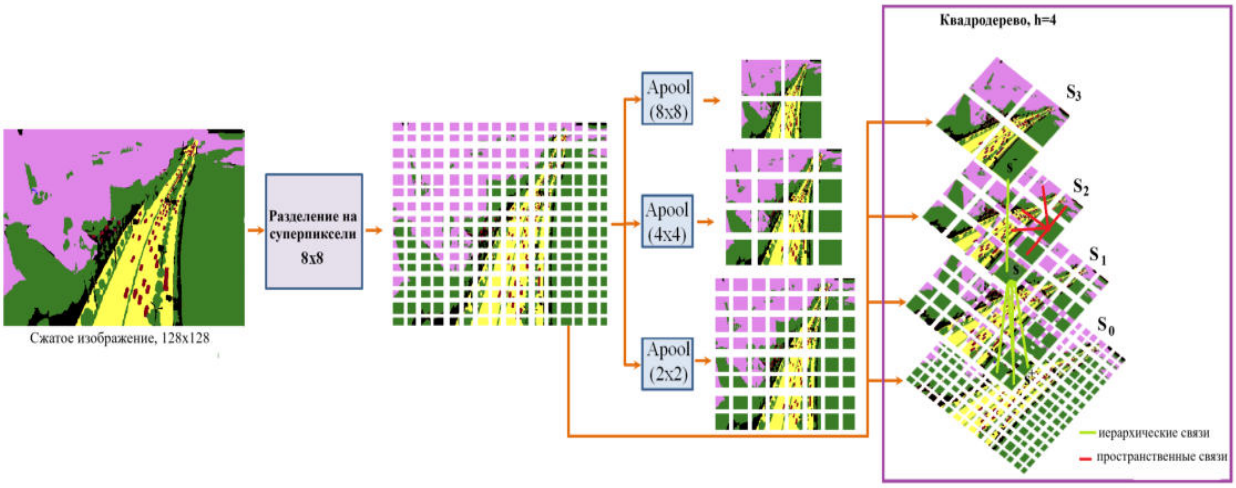

Рисунок 3.4 — Схема построения квадродерева высоты $h = 4$ в архитектуре FN-QiGSAN

3.3.1 Разделение на суперпиксели и формирование квадродерева

В графовой сети $G(\cdot)$, входящей в FN-QiGSAN, входные данные внутренних признаков изображения X^k_{cat} с числом каналов N_{ch} вначале обрабатываются с помощью $d \in \mathbb{N}$ блоков сжатия, состоящих из слоев усредняющего пулинга с полем $q \times q$, $q = 1; 2$, и двумерной свертки с ядром 3×3 (см. оранжевые стрелки на рисунке 3.3). Сформированные глобальные признаки $\mathbf{x_Q}$ в q^d раз меньше исходного снимка: величина d определяется размером входного изображения, а также выбором значений h и размера суперпикселей.

Одновременно с глобальными признаками $\mathbf{x_Q}$ формируются вспомогательные и локальные признаки – векторы $\mathbf{x_{Sob}}$ и $\mathbf{x_{local}}$ (синие и фиолетовые стрелки на рисунке 3.3). Локальные признаки $\mathbf{x_{local}}$ формируются отдельно от глобальных: для сохранения мелких деталей к ним применяется на одно преобразование пулинга с полем $q \times q$ меньше. Вспомогательные признаки $\mathbf{x_{Sob}}$ используются только в задаче многоклассовой сегментации для выделения глобально неоднородных участков исходного снимка и границ между ними. Признаки $\mathbf{x_{Sob}}$ формируются в результате фильтрации по Собелу [215]: для каждого пикселя вычисляется градиент яркости по горизонтали и вертикали.

Для обработки изображений высокого разрешения с помощью графовых архитектур из-за квадратичной вычислительной сложности умножения на матрицу смежности их часто разбивают на подобласти-суперпиксели нерегулярной или регулярной структуры для уменьшения числа элементов, вобрабатываемых с помощью матрицы. В первом случае пиксели группируются в суперпиксели в

соответствии с их яркостными свойствами [216] с помощью специальных алгоритмов, например байесовской адаптивной сегментации [217]. Во втором случае все суперпиксели имеют заданный одинаковый размер и форму.

Структура пространственно-иерархического квадродерева в полной мере позволяет реализовать совместную обработку разномасштабных элементов изображения. Пример построения квадродерева высоты $h = 4$ представлен на рисунке 3.4. Три вектора признаков разбиваются на суперпиксели, после чего на их основе формируются многомасштабные представления, соответствующие структуре квадродерева с числом слоев h: векторы $\mathbf{x}_{\mathbf{Q}}^{(\mathbf{sp})}$, $\mathbf{x}_{\mathbf{Sob}}^{(\mathbf{sp})}$ (в многоклассовом случае) и $\mathbf{x}_{\mathbf{local}}^{(\mathbf{sp})}$. Слой S_0 квадродерева формируется из исходных векторов признаков, а для построения слоев $S_1, \ldots, S_{h-1}$ используется усредняющий пулинг $\mathbf{x}_{\mathbf{Q}}^{(\mathbf{sp})} = \left(avg(\mathbf{x}^{(\mathbf{sp})}), \mathbf{x}^{(\mathbf{sp})}\right) \in \mathbb{R}^{N\times 1}$, где $\mathbf{x}^{(sp)}$ – вектор признаков исходного изображения после применения первого преобразования уменьшения размерности, разбитого на суперпиксели, N – общее количество суперпикселей, и $avg(\mathbf{x}^{(\mathbf{sp})}) = \left(avg_h(\mathbf{x}^{(\mathbf{sp})}), \ldots, avg_2(\mathbf{x}^{(\mathbf{sp})})\right)$, и $avg_p(\mathbf{x}) = avg\_pool(\mathbf{x}, 2^p \times 2^p)$, $p = \overline{1,h-1}$ – размер области пулинга по среднему значению.

3.3.2 Графовый блок и формирование выходного изображения

В графовом блоке обработка мультимасштабных разбиений изображения реализована в отдельных ветвях (см. рис. 3.3, блоки внутри- и меж-суперпиксельной свертки). Локальные признаки преобразуются стандартным линейным графово-сверточным слоем по графу локальных взаимосвязей в форме двумерной решетки размера $M \times M$, состоящей из M^2 пикселей в каждом суперпикселе. Результаты обработки представимы в форме:

$$\mathbf{x}_{\mathbf{lRes}}^{(\mathbf{sp})} = \mathbf{x}_{\mathbf{local}}^{(\mathbf{sp})} V \cdot B, \tag{3.28}$$

Обработка глобальных признаков изображения в FN-QiGSAN реализована разными способами для решения задач двухклассовой и многоклассовой сегментации. В первом случае блок глобальных признаков был спроектирован для предотвращения переобучения, поскольку дисбаланс разделяемых классов малоразмерных объектов (автомобилей, кораблей и т.д.) и фона в двухклассовом случае выражен значительно сильнее, чем в многоклассовом. Блок

содержал всего один слой графовой свертки и включал в себя несколько этапов обогащения матрицы смежности. Во второй задаче необходимость разделения пикселей изображения на большее количество классов требует точной обработки высокоуровневых представлений разных типов поверхностей, и поэтому многоклассовый блок был построен как более глубокая архитектура, включавшая несколько последовательных слоев графовой свертки.

Обработка глобальных признаков изображения при многоклассовой сегментации

Межсуперпиксельная обработка в FN-QiGSAN при многоклассовой сегментации реализуется двумя модифицированными графовыми слоями с самовниманием (GSAN) [218] и остаточными связями:

$$\mathbf{x}_{\mathbf{gRes}}^{(\mathbf{sp})} = GSAN_{res}(GSAN_{res}(\mathbf{x}_{\mathbf{Q}}^{(\mathbf{sp})})),$$

где $GSAN_{res}(\mathbf{x}) = \mathbf{x} + GSAN(\mathbf{x})$ (индекс res означает использование остаточных связей). В $GSAN_{res}(\cdot)$ из входного вектора, обработанного полносвязным слоем (индекс $Q\text{-}lin$ означает применение линейного преобразования) $\mathbf{x}_{\mathbf{Q\text{-}lin}}^{(\mathbf{sp})} = GeLU(Linear(\mathbf{x}_{\mathbf{Q}}^{(\mathbf{sp})}))$, формируется матрица внимания $X_{att} = \mathbf{x}_{\mathbf{Q\text{-}lin}}^{(\mathbf{sp})} \cdot (\mathbf{x}_{\mathbf{Q\text{-}lin}}^{(\mathbf{sp})})^T \in \mathbb{R}^{\frac{N}{M^2} \times \frac{N}{M^2}}$. На основе X_{att} формируются обогащенные матрицы смежности:

$$A_1 = sfm\left(sfm((\mathbf{b},\mathbf{b}^T) + sfm(1 - A_Q)\right) \odot X_{att}),$$
$$A_2 = sfm\left(sfm\left(C_{2d}^{3\times3}(sfm(\mathbf{x}_{\mathbf{Sob}}^{(\mathbf{sp})} \cdot (\mathbf{x}_{\mathbf{Sob}}^{(\mathbf{sp})})^T) + sfm(1 - A_Q)\right) \odot X_{att})\right),$$

где $\mathbf{b} \in \mathbb{R}^{\frac{N}{M^2} \times 1}$ – обучаемый вектор весов, $C_{2d}^{3\times3}$ – обработка двумя слоями двумерной свертки с размером ядра 3, а $\odot$ – операция поэлементного произведения.

Обработка глобальных признаков изображения при двухклассовой сегментации

Для меж-суперпиксельной обработки при двухклассовой сегментации используется модифицированный GSAN слой, который получает на вход вектор

$\mathbf{x}_Q^{(sp)}$ и обрабатывает его полносвязным слоем $\mathbf{x}_{Q\text{-}lin}^{(sp)} = GeLU(Linear(\mathbf{x}_Q^{(sp)})) \in \mathbb{R}^{\frac{N}{M^2} \times M^2}$. Далее в блоке производится обогащение матрицы смежности квадродерева A_Q с помощью матрицы внимания $X_{att} = \mathbf{x}_{Q\text{-}lin}^{(sp)} \cdot (\mathbf{x}_{Q\text{-}lin}^{(sp)})^T$, $X_{att} \in \mathbb{R}^{\frac{N}{M^2} \times \frac{N}{M^2}}$. Из A_Q формируются матрицы A_i, $i = \overline{0,2}$ одним из двух способов (номер способа являлся гиперпараметром при обучении модели):

$$\begin{aligned} & A_0 = A_Q \odot X_{att}, && A_0 = A_Q \odot sfm(X_{att}), \\ \text{I} : & A_1 = C_{2d}^{3\times 3}(X_{att} - A_0), && \text{II} : A_1 = sfm(A_Q) \odot X_{att}, \\ & A_2 = 0, && A_2 = sfm(A_Q \odot X_{att}). \end{aligned}$$

A_0 соответствует базовой матрице графового внимания, а матрицы A_1 и A_2 позволяют учесть признаки слабой интенсивности. Затем матрицы A_0 и A_1 объединяются в матрицу A_3 и нормируются одним из четырех способов (номер способа являлся гиперпараметром при обучении модели):

$$A_3 = \begin{aligned} & \text{I} : A_0 + sfm(A_1) \cdot \sigma(sfm(A_1) - \delta), \\ & \text{II} : (sfm(A_1) + sfm(A_0))\sigma(sfm(A_1) + sfm(A_0) - \delta), \\ & \text{III} : sfm(A_0 + A_1 \cdot \sigma(A_1 - \delta)), \\ & \text{IV} : sfm(A_1 + A_0 \cdot \sigma(A_0 - \delta)), \end{aligned}$$

где $\delta \in [0,0.5]$ – порог значимости признака, а $\sigma(\cdot)$ – сигмоидная функция активации. После выполняется слияние признаков матриц A, A_3, A_2 ($a_i, i = \overline{1,3}$ – обучаемые веса), и на основе объединенной матрицы вычисляется вектор $\mathbf{x}_{fused}^{(sp)}$. Наконец, для ранжирования признаков пикселей в составе $\mathbf{x}_{fused}^{(sp)}$, тот приводится к размерности $\frac{N}{M^2} \times M^2 \times k$, $k \in \mathbb{N}$, после чего к нему применяется функция $\text{softmax}(\cdot) = sfm(\cdot)$, ранжирующая значения по k. Далее формируется выходной вектор блока $\mathbf{x}_{\mathbf{gRes}}^{(\mathbf{sp})}$.

$$\begin{aligned} \mathbf{x}_{fused}^{(sp)} &= (a_1 \cdot A_3 + a_2 \cdot A_2 + a_3 \cdot A_Q) \cdot \mathbf{x}_Q^{(sp)}, \\ \mathbf{x}_{\mathbf{gRes}}^{(\mathbf{sp})} &= \mathbf{x}_{Q-lin}^{(sp)} + GeLU(B \odot sfm(\mathbf{x}_{fused}^{(sp)})) \end{aligned}$$

Объединение глобальных и локальных признаков

После обработки в графовом блоке глобальные и локальные признаки $\mathbf{x}_{\mathbf{gRes}}^{(\mathbf{sp})}$ и $\mathbf{x}_{\mathbf{lRes}}^{(\mathbf{sp})}$ объединяются. Для этого каждый вектор разделяется на h под-

векторов $\mathbf{x_{gRes}}^{(i)}$ и $\mathbf{x}_{lRes}^{(i)}$, $i = \overline{1,h}$, в соответствии с принадлежностью элементов к соответствующим слоям квадродерева. Затем эти векторы конкатенируются в соответствии с масштабом: поскольку локальные признаки при $q = 2$ могут иметь масштаб больший в два раза, чем у глобальных признаков:

$$\mathbf{x_{cat}}^{(i)} = \text{concat}(\mathbf{x_{lResEx}}^{(i)}, \mathbf{x_{gResEx}}^{(i)}),\ i = \overline{1,h+1},$$

где $\mathbf{x_{lResEx}} = \left(0_{\frac{N^2}{2^h M^2} \times M^2}, \mathbf{x}_{lRes}\right)$ и $\mathbf{x_{gResEx}} = \left(\mathbf{x_{gRes}}, 0_{\frac{N^2}{M^2} \times M^2}\right)$ – дополненные нулями векторы результатов обработки глобальных и локальных признаков в графовом блоке. Дополнение может быть реализовано за счет умножения на матрицы I_{loc} и I_{glob}, заданные в разделе 3.2.2.

Если же $q = 1$, то векторы глобальных и локальных признаков имеют одинаковый масштаб, и $I_{loc} = \left(I_{\frac{N^2}{M^2} \times \frac{N^2}{M^2}}\right)$ и $I_{glob} = \left(I_{\frac{N^2}{M^2} \times \frac{N^2}{M^2}}\right)$. Объединенные признаки обрабатываются с помощью сверточных слоев с функцией активации $GeLU$.

3.4 Результаты сегментации изображений

3.4.1 Описание тестируемых наборов данных

В разделе представлены результаты сегментации аэрокосмических изображений высокого разрешения с помощью информированной архитектуры FN-QiGSAN. Были рассмотрены четыре открытых набора (см. табл. 33) снимков, полученных с помощью спутников и БПЛА: HRSID [219], SSDD [220], UAVid [221] и UDD [222]. Первые два набора содержат радиолокационные изображения кораблей в море. Наборы UAVid и UDD содержат RGB-снимки сложных городских ландшафтов. Общим у четырех наборов является выраженный дисбаланс разделяемых классов – доля пикселей, соответствующих малым объектам на изображениях (автомобили, корабли), во всех случаях не превосходит 2.5%.

Рассматривались две задачи – обнаружение малоразмерных объектов (двухклассовая сегментация) и полная (многоклассовая) сегментация изображения. В первом случае целевым классом в наборах HRSID и SSDD являлись

Таблица 33 — Описание тестируемых наборов данных

Набор	Размер набора	Разрешение	Высота, пикс.	Ширина, пикс.	Малые объекты, %
HRSID	1962	0.5 – 3 м/пикс.	800	800	0.41
SSDD	1160	1 – 10 м/пикс.	190 – 526	214 – 668	1.25
UAVid	200	Ultra HD 4K	2160	3840 – 4096	2.5
UDD	120	Ultra HD 4K	2160	3840 – 4096	0.87

корабли, а в UAVid и UDD – движущиеся и припаркованные автомобили. Все остальные поверхности классифицировались как фон.

В задаче многоклассовой сегментации рассматривались только наборы UDD и UAVid, поскольку для радиолокационных снимков не было соответствующей разметки. В UDD выделялось пять классов: «Автомобили», «Дороги», «Здания», «Растительность» и «Шум» (в исходном наборе «Clutter»). В наборе UAVid рассматривались 5 классов из содержащихся в нем восьми: похожие классы были объединены между собой, например, «Деревья» и «Растительность», «Движущиеся автомобили» и «Припаркованные автомобили». Сделано это потому, что для обработки более сложного и несбалансированного набора UDD требовалось предобучение базового кодировщика на схожем по наблюдаемым объектам наборе, в качестве которого использовался UAVid.

3.4.2 Гиперпараметры

К обучающим данным применялись случайные повороты (от 1 до 180 градусов) и сдвиги (от 1 до 128 пикселей) по горизонтали и вертикали. Для повышения доли пикселей целевых объектов (см. табл. 33) в наборах HRSID и UDD фрагменты, содержащие корабли или машины, добавлялись в обучающий набор с повторением, притом к ним применялись преобразования случайного изменения контраста и размытия по Гауссу [223]. Точность сегментации оценивалась с помощью классической метрики $F_1 = \frac{2 \cdot \text{Точность} \cdot \text{Полнота}}{\text{Точность} + \text{Полнота}}$.

В двухкласовой задаче из-за малых масштабов целевых объектов размер обрабатываемого изображения составлял 256×256, а в качестве функции потерь использовалась стандартная кросс-энтропия. Точность обработки каждого набора оценивалась по результатам пятикратной кросс-проверки. Для обучения использовались 70% выбранных для кросс-валидации обучающих фолдов,

а оставшиеся 30% использовались как валидационное множество (общая доля данных для обучения не превосходила 56% от общего количества). Дополнительно в двухклассовой задаче на примере наборов SSDD и HRSID исследовался вопрос об эффективности FN-QiGSAN при обработке малых по числу элементов наборов [43]. Поэтому в HRSID рассматривалось только множество, выделенное в исходном наборе как тестовое и содержащее меньшее количество элементов.

В задаче многоклассовой сегментации размер обрабатываемого снимка составлял 512×512 из-за необходимости обрабатывать помимо автомобилей крупные объекты. Для обучения моделей использовалась функция потерь вида:

$$Loss(\mathbf{x}) = CrossEntropy(\boldsymbol{\omega}_1, \mathbf{x}, \mathbf{y}) + \boldsymbol{\alpha} \cdot DiceLoss(\boldsymbol{\omega}_2, \mathbf{x}, \mathbf{y}),$$

где первое слагаемое – взвешенная кросс-энтропия, а второе соответствует мере перекрытия классов [224], которая используется для более точной обработки границ объектов в случае дисбаланса классов. Оба слагаемых используют взвешивание: классам или приписываются равные веса, или классу малых объектов («Автомобили») придается больший вес. Гиперпараметры, используемые при обучении сети и построении функции потерь, представлены в таблице 34. Обучение производилось одновременно на двух картах NVIDIA A100.

Таблица 34 — Гиперпараметры обучения моделей

Параметр	**Описание**	**Диапазон изменения**
h	Высота квадродерева	4; 5
M	Размер суперпикселя	8; 16
d	Количество блоков сжатия	1; 2
q	Размер поля пулинга	1; 2
$\boldsymbol{\omega}_i$, $i = 1{,}2$	Вектор весов для функции потерь	$(0.33; 0.16; 0.16; 0.16; 0.16)$; $(0.2; 0.2; 0.2; 0.2; 0.2)$
Pre-train	Предобученные веса для базовых моделей	нет; веса из репозитория PyTorch для DeepLabV3 и FCN; наборы Imagenet, ADE20K
$\boldsymbol{\alpha}$	Коэффициент значимости DiceLoss	0; 1
batch	Размер обучающего батча	16; 32
opt	Используемый оптимизатор	Adam; AdamW

3.4.3 Нейросетевые архитектуры для сравнения с FN-QiGSAN

FN-QiGSAN сравнивается с рядом популярных НС для сегментации изображений, включая сверточные DeepLabV3 [27] на базе MobileNetV3, ENet [139], сверточная сеть на базе ResNet50 (FCN) [225] на базе ResNet50, PSPNet [140] (пирамидальная сеть), U-Net [196] и трансформерные архитектуры SegFormer [141], DPT и LWGANet [36]. LWGANet – новая архитектура 2025 года на базе UNetFormer, разработанная специально для обработки аэрокосмических изображений, включая БПЛА-снимки, демонстрирующая на UAVid лучшие результаты точности сегментации среди всех известных моделей (англ. state-of-the-art, SOTA). Кроме того, в задаче двухклассовой сегментации также рассматривались модификации DeepLabV3 и U-Net++ с механизмом детализированного внимания (англ. across feature map attention, AFMA) [226], которые были специально разработаны для повышения точности выделения малых объектов.

Перечисленные архитектуры (за исключением AFMA, который был интересен только в контексте сравнения с альтернативными методами обработки малых объектов) использовались как варианты реализации кодировщика $F(\cdot)$ в архитектуре FN-QiGSAN. Число эпох обучения во всех случаях составляло от 120 до 180. Для набора UDD из-за его более выраженной несбалансированности и изменчивости, сегментаторы дополнительно предобучались на UAVid около 120 эпох для настройки на особенности предметной области. Без этого базовые модели нередко теряли целые классы объектов.

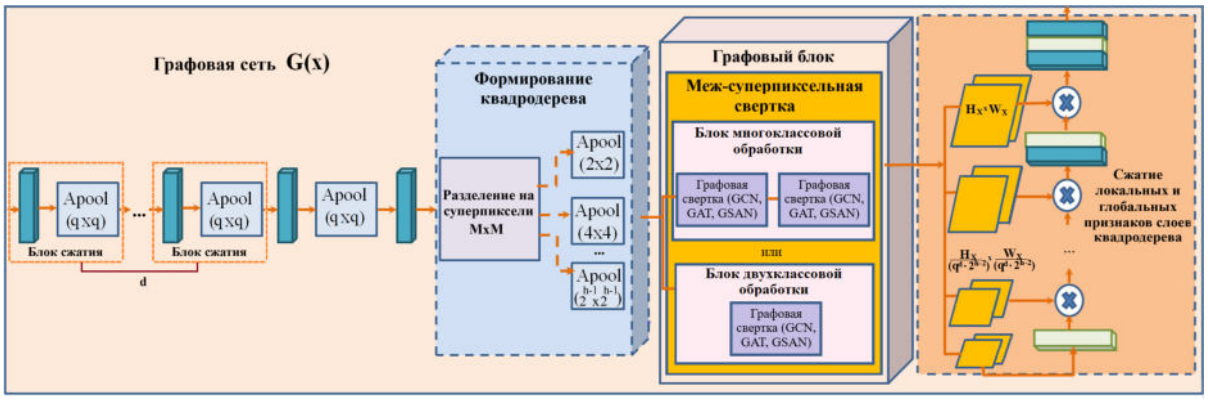

Рисунок 3.5 — Архитектуры альтернативных реализаций $G(\cdot)$: QiGSAN, QiGCN, QiGAT и GSAN, GCN, GAT

Помимо разных базовых сетей, также рассматривались альтернативные подходы к построению графовой сети $G(\cdot)$ (см. рис. 3.5), притом как

информированные квадродеревом, так и неинформированные. Многие современные графовые архитектуры, применяющиеся для обработки изображений, и которые использовались для сравнения с информированным двухветочным блоком, являются модификациями базовых графовых блоков: простого линейного [227] (см. формулу (2.9)) или блока с вниманием (GAT) [228]. Например, GraphSAGE [229] отличается от линейного графового блока только тем, что агрегация характеристик узлов выполняется по предопределенному количеству соседей. Существуют также модификации GAT, которые реализуют механизмы самовнимания для обработки объединённых признаков узлов [230; 231] или для определения глобальных атрибутов узлов [218].

Все альтернативные архитектуры не включали ветви обработки локальных признаков снимка (синие стрелки внизу на рис. 3.3), а при многоклассовой сегментации – еще и ветви обработки вспомогательных признаков (лиловые стрелки). Графовый блок в альтернативных архитектурах реализован как последовательность из одного или двух (по числу умножений на матрицу смежности в FN-QiGSAN) стандартных графово-сверточных слоев [227] и графовых сетей с вниманием [228] или самовниманием [218].

В информированных архитектурах свертка выполнялась по графу квадродереву. Эти сети обозначены как QiGSAN, QiGCN, QiGAT – в соответствии с реализацией графовой свертки. В неинформированных архитектурах графовая свертка производилась по графу двумерной решетке, который не включал в себя многомасштабные признаки изображения. В неинформированных сетях блоки формирования и сжатия слоев квадродерева, выделенные пунктирными линиями на рис. 3.5 (в центре и справа), отсутствовали. Дополнительно при многоклассовой сегментации была протестирована модификация FN-QiGSAN, использовавшая обычные GSAN-слои для обработки глобальных признаков – MfQiGSAN (Modified QiGSAN), для оценки эффективности модифицирования (см. формулу (3.3.2)) матриц смежности.

3.4.4 Результаты обработки изображений в задаче двухклассовой сегментации

В таблице 35 представлены наилучшие для каждой тестируемой архитектуры значения метрики F_1, полученные в ходе перекрестной проверки на пяти фолдах для наборов HRSID, SSDD, UAVid и UDD (максимальные для каждого датасета значения выделены жирным шрифтом). Приведены результаты обработки сверточными сетями (ENet, DeepLabV3, FCN и др.), трансформерными архитектурами (SegFormer, LWGANet) и графовыми ансамблями QiGCN, QiGSAN, QiGAT, GCN, GSAN и GAT. Более подробный анализ результатов, включая оценки точности, полученные для каждого кодировщика $F(\cdot)$ в рамках ансамбля, представлен в дальнейших подразделах.

FN-QiGSAN повышает точность сегментации кораблей в сравнении со всеми протестированными архитектурами. Прирост средних значений F_1-меры составляет 4.16-66.24% в сравнении с результатами трансформерных сетей (до 39.28% между лучшими конфигурациями). Для сверточных архитектур прирост средних значений F_1-меры составляет 5.37-62.05% (до 25.57% между лучшими конфигурациями).

Также FN-QiGSAN повышает точность сегментации автомобилей на БПЛА-изображениях. Прирост средних значений F_1-меры составляет 20.88-32.81% в сравнении с результатами трансформерных сетей (до 28.61% между лучшими конфигурациями), и 7.78-23.76% (до 8.48% между лучшими конфигурациями) – для сверточных архитектур.

FN-QiGSAN демонстрирует результаты максимальной точности в сравнении со всеми графовыми ансамблями. Прирост средних значений F_1-меры относительно информированных ансамблей (QiGCN, QiGAT, QiGSAN) составляет 0.54-5.28%, тогда как относительно неинформированных модификаций прирост составляет 1.45–26.90%. FN-QiGSAN также превосходит AFMA DeepLabV3 и AFMA U-Net++: прирост средних значений F_1-меры составляет от 15.09% до 57.45%.

Таблица 35 — Средние и медианные значения F_1-меры ($\times 100\%$), полученные в ходе перекрестной проверки на пяти фолдах для наборов HRSID, SSDD, UAVid и UDD

НС	HRSID	SSDD	UAVid	UDD
ENet	73.77 ± 9.59 (77.74)	33.04 ± 19.4 (29.28)	65.81 ± 4.14 (66.03)	59.81 ± 9.08 (63.72)
DeepLabV3	60.45 ± 18.67 (69.97)	58.35 ± 4.47 (57.81)	67.99 ± 4.79 (68.57)	49.08 ± 24.82 (56.05)
FCN	59.87 ± 32.15 (74.13)	21.87 ± 5.35 (21.86)	70.05 ± 2.20 (70.74)	47.85 ± 6.68 (48.24)
U-Net	74.61 ± 10.23 (79.05)	31.14 ± 11.71 (32.95)	65.17 ± 3.65 (65.76)	43.67 ± 2.56 (45.15)
PSPNet	67.73 ± 9.75 (70.18)	46.15 ± 12.66 (51.31)	67.50 ± 2.36 (67.34)	54.89 ± 5.47 (56.12)
SegFormer	74.98 ± 16.59 (82.61)	44.64 ± 16.05 (50.36)	53.31 ± 3.99 (53.25)	34.78 ± 4.19 (34.71)
LWGANet	73.36 ± 13.58 (78.34)	17.68 ± 3.29 (16.82)	57.65± 5.78 (58.70)	38.98 ± 8.37 (42.67)
Специализированные модели для выделения малых объектов				
AFMA DeepLabV3	64.05 ± 14.92 (69.14)	26.47 ± 6.49 (27.22)	59.70 ± 5.66 (61.80)	51.04 ± 3.90 (51.87)
AFMA U-Net++	62.06 ± 10.43 (66.12)	40.04 ± 6.31 (39.29)	57.52 ± 7.40 (57.50)	49.59 ± 5.93 (49.72)
Ансамбли с разными типами графовых сетей $G(\cdot)$				
GCN	73.84 ± 9.56 (76.92)	76.09 ± 3.48 (75.55)	77.08 ± 2.80 (77.13)	64.92 ± 8.64 (66.15)
GAT	68.08 ± 9.02 (71.54)	74.87 ± 4.01 (76.58)	76.49 ± 2.40 (76.34)	40.69 ± 12.59 (34.23)
GSAN	69.38 ± 13.55 (72.22)	78.71 ± 2.36 (78.38)	70.62 ± 8.29 (72.25)	57.00 ± 8.80 (57.60)
QiGCN	75.27 ± 10.00 (78.72)	78.64 ± 4.27 (77.98)	77.32 ± 2.90 (77.25)	66.45 ± 7.91 (66.76)
QiGAT	76.53 ± 10.86 (80.36)	79.83 ± 7.41 (81.79)	74.28 ± 3.50 (74.22)	55.71 ± 4.18 (55.74)
QiGSAN	78.60 ± 11.38 (82.87)	78.71 ± 0.73 (78.89)	76.04 ± 3.74 (77.11)	60.88 ± 4.62 (62.42)
FN-QiGSAN	**79.14 ± 11.19 (83.56)**	**83.92 ± 5.76 (84.66)**	**78.53 ± 2.76 (78.57)**	**67.59 ± 7.63 (68.40)**

Результаты сегментации кораблей на спутниковых изображениях

В таблицах 3.6 для наборов HRSID и SSDD представлены оценки точности сегментации кораблей, включающие средние значения метрики F_1, полученные

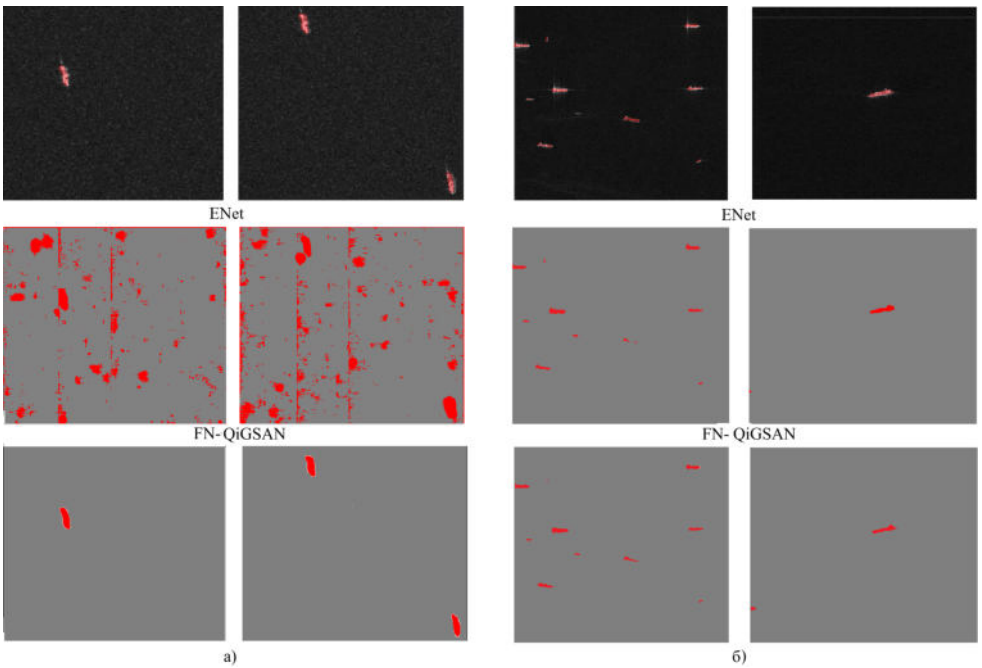

Рисунок 3.6 — Сегментация изображений из наборов HRSID (а) и SSDD (б) ENet и FN-QiGSAN

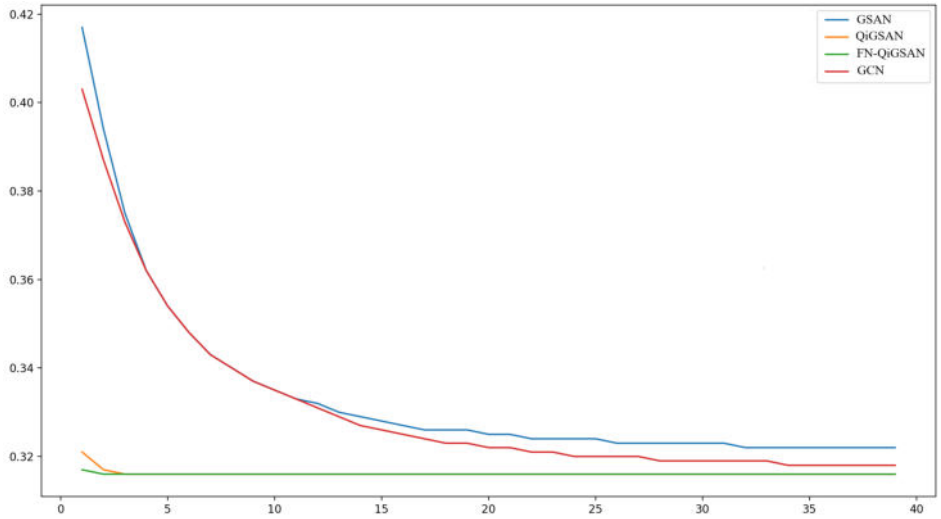

Рисунок 3.7 — Изменение функции потерь при обучении на наборе HRSID сетей GSAN, GCN, QiGSAN и FN-QiGSAN

в ходе перекрестной проверки на пяти фолдах, их среднее квадратичное отклонение и медиану (в скобках). Лучшие значения для каждого кодировщика выделены жирным шрифтом. Примеры сегментируемых изображений и результатов их обработки представлены на рисунке 3.6.

FN-QiGSAN во всех случаях повышает точность сегментации кораблей в сравнении с результатами базовых сверточных сетей. Для HRSID прирост средних значений составляет 2.01-17.85% (в среднем – 7.33%), а для SSDD – 12.99-59.99% (в среднем 41.55%). FN-QiGSAN стабильнее альтернативных гра-

фовых ансамблей, часть из которых более несбалансированном наборе HRSID не улучшают результат сверточной сети. Точность, получаемая FN-QiGSAN, в среднем на 0.54-16.21% (в среднем 4.82%) выше, чем при обработке информированными архитектурами QiGCN, QiGAT и QiGSAN, и на 3.57-20.42% (в среднем 7.83%) – чем при обработке архитектурами без информирования GCN, GAT и GSAN.

FN-QiGSAN также обучается быстрее рассмотренных графовых ансамблей – пример графиков функций потерь представлен на рисунке 3.7. Можно заметить, что информированные квадродеревом сети обучаются быстрее, чем графовые архитектуры, использующие свертку по двумерной решетке. Этот результат подтверждает утверждения 8-10 из раздела 3.2.

Таблица 36 — Средние и медианные значения F_1-меры (×100%), полученные в ходе перекрестной проверки на пяти фолдах для набора HRSID

	Базовая НС	**GCN**	**GAT**	**GSAN**	**QiGCN**	**QiGAT**	**QiGSAN**	**FN-QiGSAN**
ENet	73.77± 9.59 (77.74)	61.57 ± 17.39 (64.86)	11.77 ± 10.29 (9.65)	56.40 ± 5.09 (58.40)	71.97 ± 9.01 (75.38)	72.68± 6.79 (74.81)	75.66 ± 14.57 (79.62)	**77.85± 10.34 (81.80)**
DeepLabV3	60.45 ± 18.67 (69.97)	66.28 ± 13.9 (74.44)	31.46 ± 19.62 (38.41)	50.91 ± 19.68 (63.00)	67.23 ± 14.69 (75.32)	44.18 ± 28.9 (51.15)	70.03 ± 14.1 (76.97)	**72.83± 15.21 (81.41)**
PSPNet	67.73 ± 9.75 (70.18)	60.39 ± 12.03 (61.36)	12.27 ± 13.46 (8.34)	53.71 ± 8.81 (55.70)	60.93 ± 13.77 (62.32)	18.96 ± 32.85 (0.0)	65.50 ± 8.86 (63.89)	**74.56± 10.81 (79.69)**
FCN	59.87 ± 32.15 (74.13)	50.65 ± 34.05 (65.58)	23.65 ± 21.70 (22.49)	58.15 ± 6.47 (60.09)	72.77 ± 8.16 (76.02)	72.64 ± 4.95 (74.45)	74.94 ± 7.93 (78.47)	**77.72± 8.27 (81.10)**
U-Net	74.61 ± 10.23 (79.05)	35.73 ± 41.27 (35.40)	29.54 ± 15.27 (28.31)	50.35 ± 11.80 (53.34)	66.84 ± 8.96 (68.35)	0.0 ± 0.0 (0.0)	70.84 ± 8.49 (72.91)	**76.98± 9.46 (80.87)**
SegFormer	72.77 ± 16.08 (78.19)	67.56 ± 13.97 (70.92)	52.00 ± 34.84 (67.04)	57.64 ± 19.54 (63.14)	69.18 ± 15.24 (74.21)	69.26 ± 17.39 (73.94)	71.89 ± 16.94 (77.34)	**74.78± 15.91 (80.84)**
LWGANet	73.36 ± 13.58 (78.34)	73.84 ± 9.56 (76.92)	68.08 ± 9.02 (71.54)	69.38 ± 13.55 (72.22)	75.2 ± 10.00 (78.72)	76.53 ± 10.86 (80.36)	78.60 ± 11.38 (82.87)	**79.14± 11.19 (83.56)**

Таблица 37 — Средние и медианные значения F_1-меры (×100%), полученные в ходе перекрестной проверки на пяти фолдах для набора SSDD

	Базовая НС	**GCN**	**GAT**	**GSAN**	**QiGCN**	**QiGAT**	**QiGSAN**	**FN-QiGSAN**
ENet	33.04 ± 19.43 (29.28)	76.09 ± 3.48 (75.55)	74.87 ± 4.01 (76.58)	78.71 ± 2.36 (78.38)	72.63 ± 5.78 (74.56)	75.54 ± 8.5 (80.07)	81.18 ± 2.15 (81.41)	**82.29± 2.93 (82.15)**
DeepLabV3	58.35 ± 4.47 (57.81)	65.84 ± 7.25 (65.73)	52.72 ± 5.98 (55.80)	50.92 ± 34.44 (65.25)	68.20 ± 7.51 (70.23)	67.81 ± 3.24 (68.51)	68.65 ± 5.86 (70.28)	**71.34± 6.22 (72.79)**
PSPNet	46.15 ± 12.66 (51.31)	74.32 ± 8.64 (76.31)	68.56 ± 20.55 (78.65)	78.80 ± 10.76 (81.52)	77.27 ± 5.97 (77.67)	79.83 ± 7.41 (81.79)	77.49 ± 4.85 (78.91)	**83.92± 5.76 (84.66)**
FCN	21.87 ± 5.35 (21.86)	74.61 ± 5.62 (74.56)	71.98 ± 7.63 (73.83)	74.96 ± 3.92 (73.86)	78.64 ± 4.27 (77.98)	72.71 ± 10.77 (76.82	78.35 ± 6.03 (78.09))	**81.86± 3.15 (81.19)**
U-Net	31.14 ± 11.71 (32.95)	68.52 ± 10.75 (69.94)	74.76 ± 2.89 (75.12)	75.65± 6.31 (77.43)	72.44 ± 5.80 (73.75)	75.54 ± 3.60 (75.66)	78.28 ± 3.91 (79.31)	**80.00± 3.46 (80.18)**
SegFormer	46.82 ± 15.65 (52.16)	51.93 ± 19.22 (56.99)	55.35 ± 15.10 (60.32)	59.78 ± 12.64 (60.73)	56.38 ± 19.10 (64.55)	59.37 ± 21.15 (66.83)	60.16 ± 21.67 (69.43)	**72.35± 3.70 (72.5)**
LWGANet	18.03 ± 3.39 (17.52)	73.47 ± 1.88 (73.16)	71.85 ± 8.76 (75.59)	68.75 ± 16.78 (76.32)	63.58 ± 20.33 (73.50)	77.82 ± 1.02 (77.54)	78.71 ± 0.73 (78.89)	**79.79± 0.57 (79.67)**

Результаты сегментации автомобилей на БПЛА-изображениях

В таблицах 38 и 39 для наборов UAVid и UDD представлены оценки точности сегментации автомобилей по значениям метрики F_1, полученные в ходе перекрестной проверки на пяти фолдах. Примеры сегментируемых изображений и результатов их обработки представлены на рисунке 3.8. FN-QiGSAN во всех случаях демонстрирует большую точность сегментации автомобилей в сравнении с результатами базовых сетей. Для UAVid прирост средних значений F_1-меры относительно результатов базовой сети составляет 7.58-19.36% (в среднем 10.88%), а для UDD – 6.29-16.22% (в среднем 10.37%). Средняя точность по метрике F_1, получаемая FN-QiGSAN, на 1.14-11.24% (в среднем 3.83%) выше, чем при обработке информированными ансамблями QiGCN, QiGAT и QiGSAN

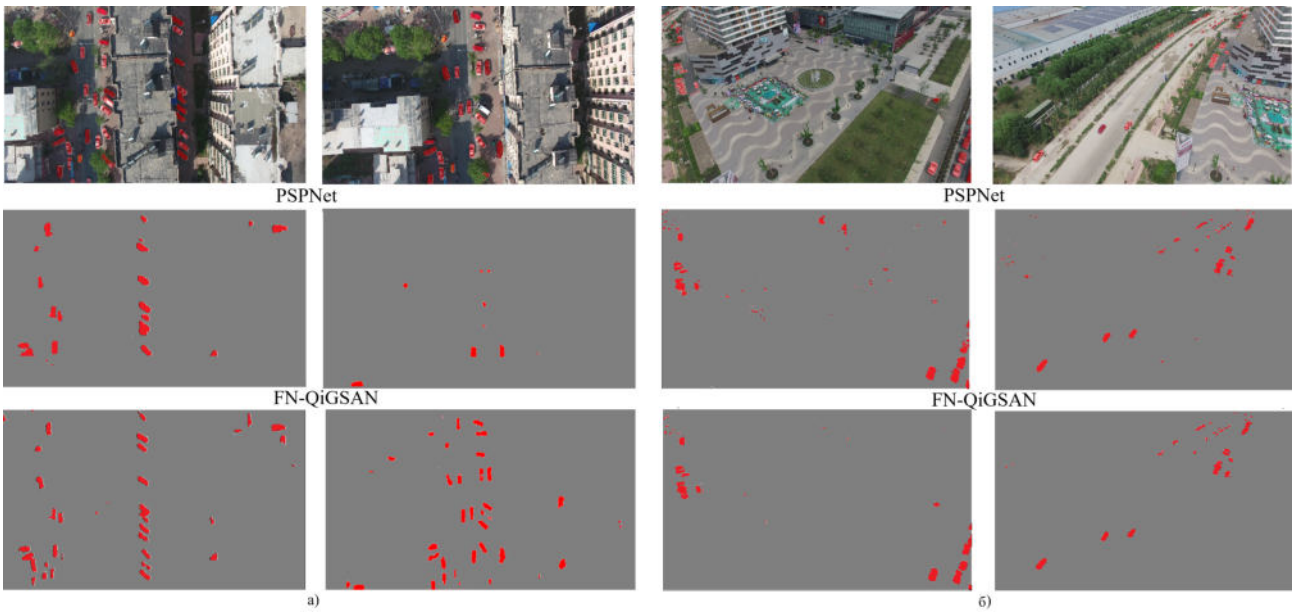

Рисунок 3.8 — Сегментация изображений из наборов UDD (а) и UAVid (б) PSPNet и FN-QiGSAN

и на 1.16-22.13% (в среднем 6.11%) – чем при обработке неинформированными GCN, GAT и GSAN. При этом в трех случаях для UAVid и в семи для UDD альтернативные неинформированные графовые ансамбли не повышают точность обработки изображений.

3.4.5 Результаты обработки изображений в задаче многоклассовой сегментации

Лучшие оценки точности сегментации наборов UAVid и UDD, полученные при многоклассовой сегментации с помощью базовых сегментаторов, а также графовыми ансамблями приведены в таблице 40. Согласно полученным значениям метрик, наилучшие результаты получены сетью FN-QiGSAN (выделены в таблице 40 полужирным шрифтом). Прирост значений F_1-меры достигается как для классов малых («Автомобили») так и крупных («Дороги», «Растительность», «Здания») объектов. В сравнении со сверточными архитектурами (DeepLabV3, U-Net, ENet, PSPNet) прирост F_1-меры для крупных и малых объектов составляет 10.72-29.83% и 5.90-34.66% соответственно. Относительно трансформеров (SegFormer, DPT, LWGANet) приросты составляют 3.72-28.38% и 11.17-48.56%, соответственно.

Таблица 38 — Средние и медианные значения F_1-меры (×100%), полученные в ходе перекрестной проверки на пяти фолдах для набора UAVid

	Базовая НС	**GCN**	**GAT**	**GSAN**	**QiGCN**	**QiGAT**	**QiGSAN**	**FN-QiGSAN**
ENet	65.81 ± 4.14 (66.03)	72.74 ± 3.47 (71.24)	68.82 ± 2.68 (68.82)	67.64 ± 6.80 (70.85)	72.85 ± 4.06 (71.08)	72.83 ± 3.94 (71.37)	72.88 ± 4.23 (71.23)	**74.36± 3.46 (72.76)**
Deeplabv3	67.99 ± 4.79 (68.57)	75.06 ± 4.10 (76.90)	75.41 ± 2.79 (76.86)	67.41 ± 6.64 (66.70)	75.66 ± 4.02 (76.53)	70.47 ± 8.57 (74.32)	76.04 ± 3.74 (77.11)	**77.88± 3.13 (78.91)**
PSPNet	67.50 ± 2.36 (67.34)	74.91 ± 4.91 (75.00)	73.40 ± 5.05 (74.72)	62.06 ± 17.38 (64.50)	74.67 ± 5.03 (74.41)	73.61 ± 3.94 (73.66)	74.46 ± 4.37 (74.17)	**76.50± 4.26 (76.28)**
FCN	70.05 ± 2.20 (70.74)	77.08 ± 2.80 (77.13)	76.49 ± 2.40 (76.34)	56.40 ± 18.29 (55.41)	77.32 ± 2.90 (77.25)	52.13 ± 30.61 (65.4)	74.09 ± 8.59 (76.90)	**78.53± 2.76 (78.57)**
U-Net	65.17 ± 3.65 (65.76)	73.26 ± 2.93 (73.25)	72.965 ± 3.94 (73.20)	70.62 ± 8.29 (72.25)	72.84 ± 4.49 (72.59)	74.28 ± 3.50 (74.22)	73.82 ± 4.28 (73.96)	**76.13± 3.55 (75.78)**
SegFormer	53.31 ± 3.99 (53.25)	40.48± 9.25 (42.22)	12.28± 17.09 (6.43)	14.16± 16.86 (10.59)	40.68± 9.06 (38.91)	57.79± 2.93 (57.09)	58.26± 2.30 (57.51)	**60.89± 2.56 (60.75)**
LWGANet	57.65± 5.78 (58.70)	65.39± 5.40 (67.10)	63.41± 3.72 (64.26)	62.42± 16.23 (69.25)	65.77± 4.95 (66.93)	72.64± 6.56 (71.02)	74.63± 5.43 (74.44)	**77.01± 3.46 (75.46)**

FN-QiGSAN демонстрирует более высокую точность сегментации, чем другие графовые ансамбли. Так, приросты составляют 0.57-9.54% и 4.94-16.4% по F_1-мере для крупных и малых объектов соответственно для сетей без информирования и 0.29-12.89% и 7.34-20.97% для информированных архитектур, в том числе и для MfQiGSAN, что демонстрирует эффективность предложенной модификации матриц смежности. FN-QiGSAN существенно превосходит альтернативные реализации графовых сетей в сложных случаях – при обработке сильно неоднородного набора UDD и при сегментации малых объектов (приросты до 20.97% и 12.89%). При обработке однородных участков (крупные объекты в UAVid) разница между моделями уменьшается.

Примеры сегментации изображений приведены на рисунке 3.9. FN-QiGSAN демонстрирует существенное улучшение точности обработки изображений в сравнении с SOTA-моделью LWGANet – для некоторых классов

Таблица 39 — Средние и медианные значения F_1-меры (×100%), полученные в ходе перекрестной проверки на пяти фолдах для набора UDD

	Базовая НС	**GCN**	**GAT**	**GSAN**	**QiGCN**	**QiGAT**	**QiGSAN**	**FN-QiGSAN**
ENet	59.81 ± 9.08 (63.72)	64.92 ± 8.64 (66.15)	3.89 ± 5.50 (0.0)	28.30 ± 26.90 (26.62)	66.45 ± 7.91 (66.76)	40.71 ± 28.81 (59.65)	60.75 ± 8.41 (60.31)	**67.59± 7.63 (68.40)**
Deeplabv3	49.08 ± 24.82 (56.05)	57.67 ± 17.62 (62.22)	17.02 ± 24.98 (4.10)	11.53 ± 23.06 (0.00)	58.74 ± 15.78 (63.62)	10.35 ± 14.63 (0.0)	58.03 ± 16.10 (60.63)	**62.25± 16.97 (67.73)**
PSPNet	54.89 ± 5.47 (56.12)	58.88 ± 4.02 (59.90)	40.69 ± 12.59 (34.23)	37.05 ± 15.80 (39.44)	60.94 ± 4.58 (62.20)	55.71 ± 4.18 (55.74)	60.88 ± 4.62 (62.42)	**62.80± 5.56 (64.13)**
FCN	47.85 ± 6.68 (48.24)	59.24 ± 3.95 (58.18)	26.26 ± 26.466 (23.95)	57.00 ± 8.80 (57.60)	59.27 ± 4.56 (58.17)	37.63 ± 26.74 (53.16)	51.54 ± 17.73 (57.22)	**60.71± 4.77 (59.30)**
U-Net	43.67 ± 2.56 (45.15)	48.80 ± 4.75 (46.98)	28.52 ± 23.06 (26.54)	19.91 ± 7.51 (19.59)	42.95 ± 12.59 (46.13)	16.93 ± 20.60 (4.53)	15.64 ± 27.09 (0.00)	**49.96± 4.83 (47.85)**
SegFormer	34.78 ± 4.19 (34.71)	31.18 ± 5.01 (32.74)	0.0 ± 0.0 (0.0)	23.11 ± 14.32 (25.42)	31.39 ± 9.19 (35.05)	40.8 ± 3.81 (40.72)	39.37 ± 4.45 (38.93)	**43.16± 3.93 (44.48)**
LWGANet	38.98 ± 8.37 (42.67)	39.08 ± 11.06 (40.54)	19.53 ± 21.60 (17.52)	37.565 ± 7.66 (33.97)	32.14 ± 18.19 (35.81)	52.69 ± 1.71 (52.78)	53.55 ± 1.30 (53.41)	**55.20± 1.28 (55.3)**

прирост точности распознавания достигает 15.11%. На снимках из набора UDD (см. рис. 3.9а) LWGANet сильно искажает область дорог и зданий в сравнении с их реальными масками классов. На снимках из набора UAVid (см. рис. 3.9б) присутствуют также искажения дорог и автомобилей. FN-QiGSAN, напротив, во всех рассмотренных примерах демонстрирует высокую точность обработки и согласованности с реальными масками классов.

Для каждого рассмотренного базового сегментатора в отдельности ансамблирование с FN-QiGSAN повышает точность сегментации крупных и малых объектов: в первом случае прирост значений F_1-меры составляет 3.29-25.04% (в среднем 10.63%), во втором – 2.05-14.87% (в среднем 11.03%). Оценки точности сегментации наборов UAVid и UDD с помощью базовых сегментаторов, ансамблей с графовыми ансамблями без информирования и FN-QiGSAN представлены на рисунке 3.10. Графовые ансамбли также повышают точность

Таблица 40 — Лучшие значения точности сегментации (метрика F_1 в %) для наборов UAVid и UDD.

Архитектура	Классы				
	Автомобили	Шум	Дорога	Растительность	Здания
	UAVid				
DeeplabV3	69.03	51.64	67.34	75.27	66.35
U-Net	63.18	48.71	67.84	85.35	78.24
ENet	55.24	50.06	65.02	86.14	77.96
PSPNet	62.44	50.18	66.18	82.61	74.84
SegFormer	66.63	55.58	71.87	87.42	83.62
DPT	41.06	46.82	63.89	84.31	76.94
LWGANet	73.9	61.34	79.17	88.32	86.68
GCN	70.239	73.52	83.74	89.89	91.91
GSAN	80.13	74.57	85.5	91.47	92.69
GAT	75.22	74.0	84.66	89.78	91.24
QiGSAN	77.73	74.91	85.94	91.08	92.11
QiGCN	71.88	73.75	83.3	89.92	91.89
QiGAT	70.18	71.44	83.78	89.81	91.71
MfQiGSAN	82.38	74.57	83.21	88.47	90.07
FN-QiGSAN	**85.07**	**75.2**	**86.59**	**92.04**	**93.15**
	UDD				
DeeplabV3	57.77	36.96	58.07	68.28	69.65
U-Net	42.14	39.28	54.11	83.15	80.34
ENet	48.49	41.76	57.75	83.0	81.16
PSPNet	52.14	40.15	55.71	79.4	79.19
SegFormer	56.77	54.02	69.57	86.79	85.25
DPT	28.23	47.31	64.67	84.88	79.86
LWGANet	63.67	43.44	62.98	82.34	82.36
GCN	60.4	54.12	72.29	89.36	88.08
GSAN	62.37	53.9	71.25	90.45	88.16
GAT	61.21	53.74	72.79	89.49	88.49
QiGSAN	55.83	54.48	70.09	90.05	88.46
QiGCN	64.27	55.36	69.75	86.14	85.93
QiGAT	62.61	50.39	67.22	83.0	84.93
MfQiGSAN	71.52	61.50	73.74	91.46	89.67
FN-QiGSAN	**76.8**	**63.28**	**78.09**	**93.88**	**91.6**

сегментации крупных объектов в среднем на 6.54% и 7.10% для информированных и неинформированных сетей, соответственно. Однако практически во всех случаях альтернативные реализации ансамбля демонстрируют снижение точности сегментации малых объектов в сравнении с базовым сегментатором: часто происходят потери класса, а среднее снижение точности достигает

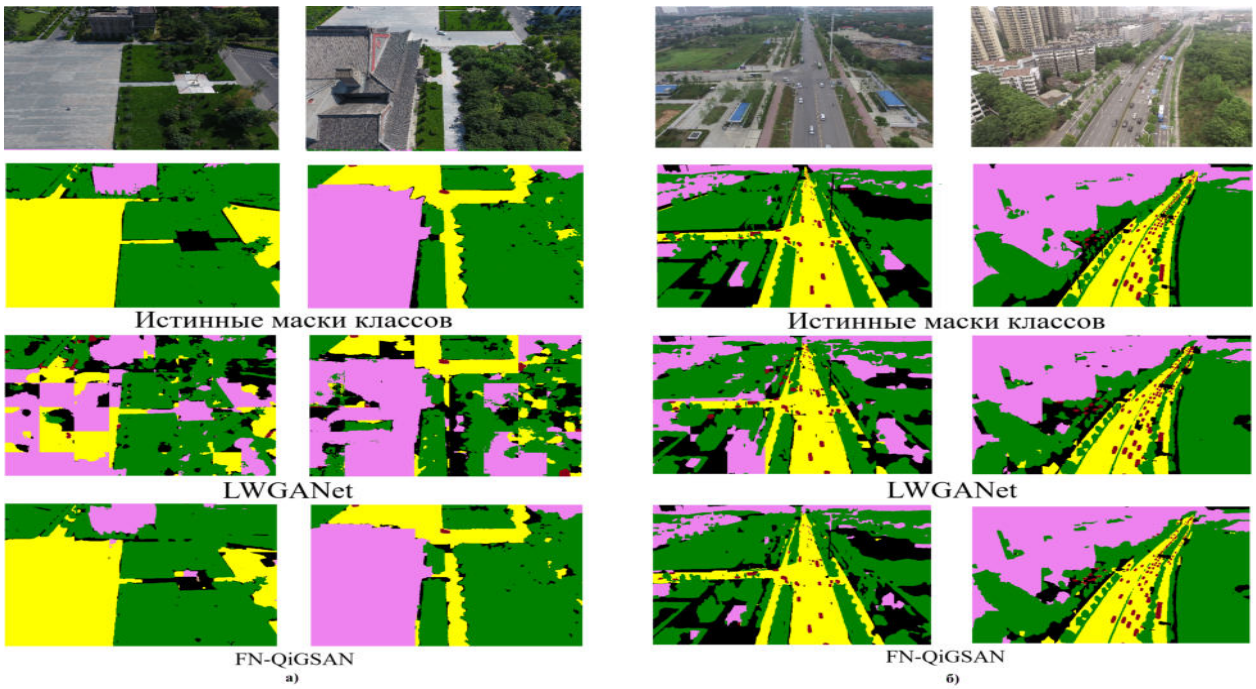

Рисунок 3.9 — Пример обработки наборов UDD (а) и UAVid (б). В столбцах: исходные изображения, маски классов (здания – сиреневого цвета, дороги – желтого, растительность – зеленого, автомобили – бордового, и шумы – черные), результаты сегментации с помощью LWGANet и FN-QiGSAN на основе LWGANet

16.05%. Данный результат демонстрирует целесообразность использования именно двухветочной архитектуры FN-QiGSAN.

3.4.6 Вычислительная эффективность FN-QiGSAN

Для задачи двухклассовой сегментации в таблице 41 для каждого набора приведены конфигурация базовой сети, давшая максимальный по величине F_1-меры результат сегментации малых объектов, а также две лучшие по точности конфигурации FN-QiGSAN, которые превосходят базовую сеть. Можно заметить, что среди выявленных конфигураций FN-QiGSAN присутствуют те, которые содержат меньшее (до 13.4 раз) число параметров, чем лучшая базовая архитектура. Исключением является только набор UDD, для которого лучшей базовой сетью была ENet, содержащая наименьшее число параметров среди всех рассмотренных архитектур. Таким образом, при двухклассовой сегмента-

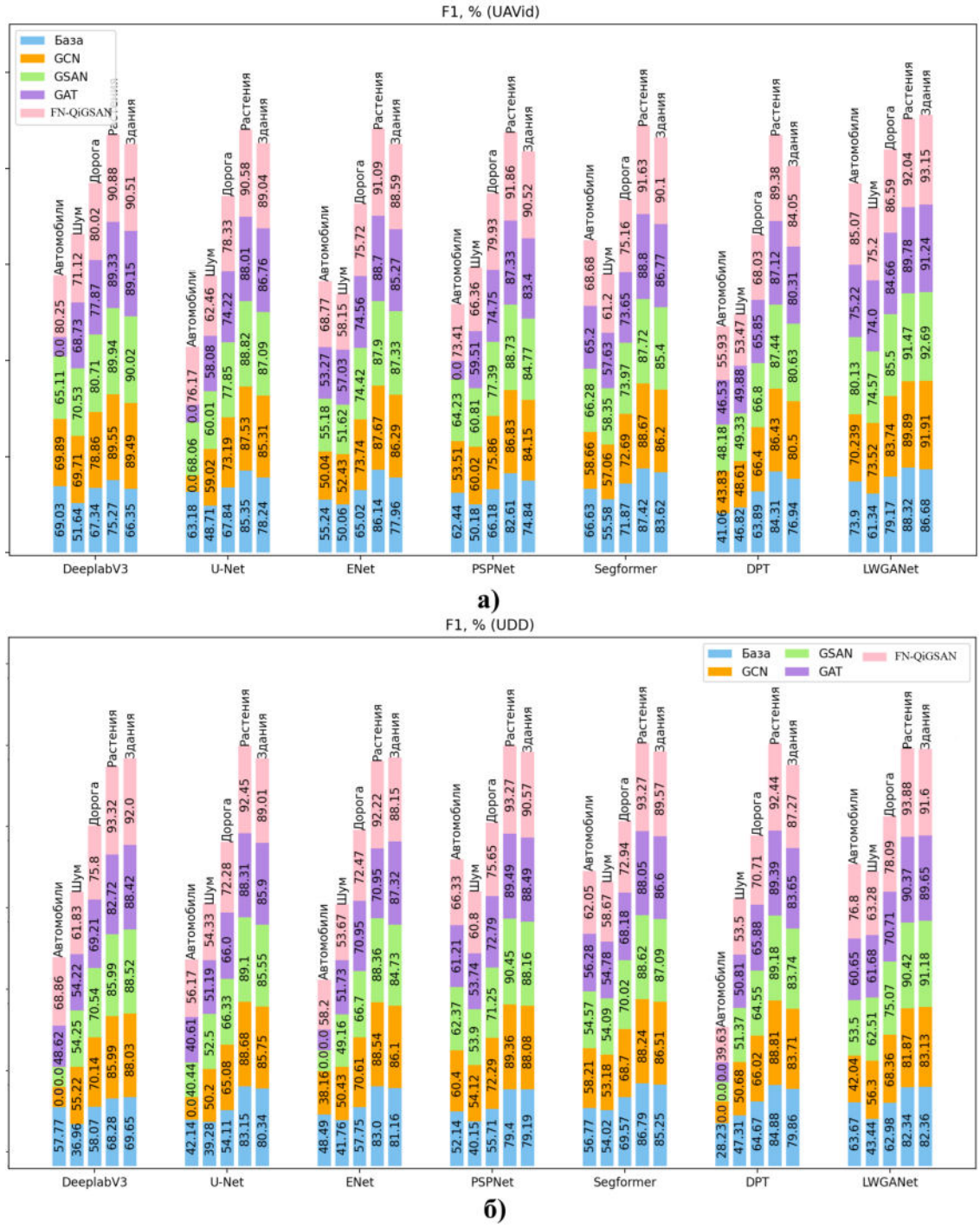

Рисунок 3.10 — Оценки точности сегментации изображений (F_1%), полученные на наборах UAVid (а) и UDD (б) FN-QiGSAN и неинформированными ансамблевыми архитектурами (GCN, GAT, GSAN) при разных базовых сегментаторах

ции FN-QiGSAN способна демонстрировать лучшие по точности результаты, чем базовая сеть, при меньшем числе параметров.

В задаче многоклассовой сегментации FN-QiGSAN демонстрирует аналогичные результаты. Сравнение лучших по точности конфигураций (для FN-QiGSAN приведены три наибольших значения, обозначенных как Top-k, $k = \overline{1,3}$) представлено на рис. 3.11. На диаграмме в верху столбца указан сред-

Таблица 41 — Лучшие по точности в задаче двухклассовой сегментации конфигурации базовых сетей и FN-QiGSAN

Набор	**Базовая сеть**	**Число параметров (база), тыс.**	**Конфигурация FN-QiGSAN (кодировщик)**	**Число параметров (FN-QiGSAN), тыс.**
HRSID	U-Net	13425	ENet, LWGANet	1089; 13314
SSDD	DeepLabV3	11029	PSPNet, ENet	1089; 49841.1
UAVid	FCN	35322	DeepLabV3, FCN	11760.1; 35953.1
UDD	ENET	358.7	ENet, DeepLabV3	1089; 11760.1

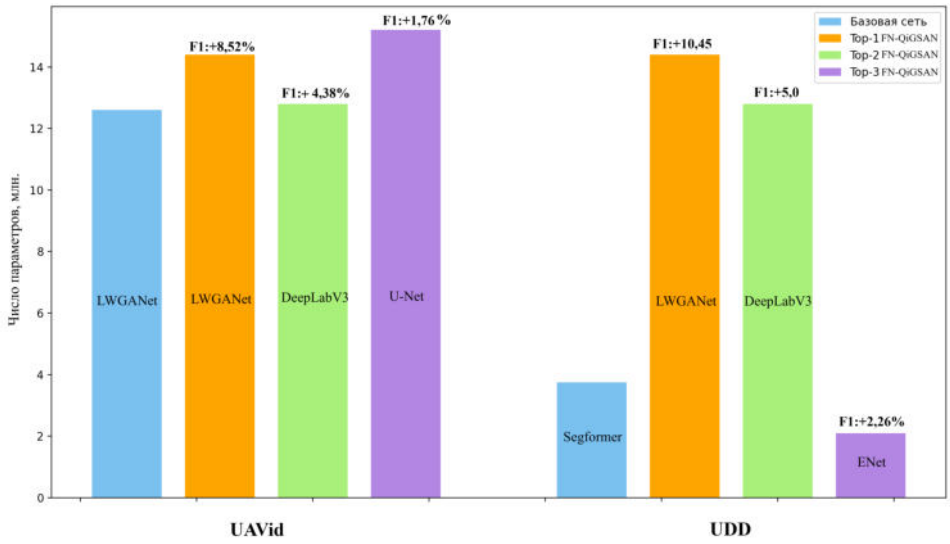

Рисунок 3.11 — Лучшие по метрике F_1 конфигурации базовых сегментаторов (их названия указаны внутри столбцов) и FN-QiGSAN для UAVid и UDD

ний прирост метрики F_1 по всем классам относительно лучших результатов, полученных базовым сегментатором. На UAVid FN-QiGSAN с базовым сегментатором DeeplabV3 превосходит LWGANet по F_1-метрике на 4.38%, а число ее параметров больше всего на 200 тыс. при том, что базовая сеть содержит порядка 12 млн. параметров. На наборе UDD FN-QiGSAN с кодировщиком ENet превосходит результаты Segformer и содержит при этом в 1.78 раз меньше параметров. Таким образом, при многоклассовой сегментации FN-QiGSAN также демонстрирует возможность получать результаты превосходящей или сопоставимой точности с помощью меньших по числу параметров сетей в сравнении с базовыми архитектурами за счет использования более легковесных кодировщиков. Это указывает на вычислительную эффективность информированной модели.

Также в обеих рассмотренных задачах FN-QiGSAN демонстрирует превосходящие по точности результаты в сравнении с информированными сетями QiGCN, QiGAT и QiGSAN. Хотя число параметров FN-QiGSAN больше – 731.1 тысяч против 370.3 у QiGCN, QiGAT и QiGSAN, основная их часть приходится на слои нормализации. Поэтому число выполняемых операций в FN-QiGSAN всего на 3.6% выше, чем остальных информированных графовых архитектурах – 181.3 против 174.84 GFLOPs соответственно, то есть вычислительная сложность сети повышается незначительно.

3.5 Выводы

В главе представлена информированная на уровне архитектуры нейросетевая модель обработки пространственных взаимосвязей между элементами изображения для обработки сильно несбалансированных наборов. Для описания глобальных связей между элементами внутренних представлений изображения, сформированных кодировщиком, применяется поле Маркова в виде квадродерева. Доказанное в главе 2 свойство эргодичности теоретически обосновывает возможность лучше обобщать закономерности, выделенные для крупных и мелких объектов в разных масштабах. Согласно теореме 6, информирование реализовано на уровне архитектуры с помощью обучаемых графово-сверточных слоев.

Представлена новая графово-сверточная ансамблевая архитектура FN-QiGSAN, информированная случайным полем Маркова для более точного описания глобальных и локальных связей в сформированных кодировщиком внутренних представлениях изображения. Аналитически и экспериментально доказано, что информированный двухветочный графовый блок, реализуемый в FN-QiGSAN, обучается более эффективно, чем сопоставимые линейные и нелинейные графовые сети, а также графовые блоки с вниманием (теоремы 9 и 10). Также доказано, что скорость обучения сетей, реализующих свертку по графу-квадродереву, выше в сравнении с традиционно применяемой в таких задачах структурой двумерной решетки (теорема 8). Данные результаты подтверждаются экспериментально.

Архитектура FN-QiGSAN была протестирована в задачах двухклассовой и многоклассовой сегментации четырех наборов аэрокосмических снимков высокого разрешения – HRSID, SSDD, UDD и UAVid. FN-QiGSAN превосходит по точности все сравниваемые с ней базовые неинформированные архитектуры в обеих задачах. Средний прирост значений F_1-меры для крупных объектов достигает 14.67% относительно трансформеров сверточных сетей соответственно, а для малых – 11.89%.

FN-QiGSAN превосходит по точности SOTA-модель для сегментации аэрокосмических снимков LWGANet – для некоторых классов прирост F_1-меры достигает 15.11%. В задаче двухклассовой сегментации прирост точности сегментации кораблей FN-QiGSAN по средним значениям F_1-меры достигает 66.24%, а автомобилей на БПЛА-изображениях – 32.81% в сравнении с трансформерными и сверточными сетями.

FN-QiGSAN демонстрирует большую вычислительную эффективность. Лучшие по точности конфигурации FN-QiGSAN демонстрирует результаты превосходящей или сопоставимой с трансформерами (SegFormer и LWGANet) точности при использовании простых сверточных кодировщиков, например U-Net, ENet или DeeplabV3, а уменьшение числа параметров сети достигает 1.78 раз в задаче многоклассовой сегментации, и 13.4 раз – в двухклассовой.

Заключение

В диссертации предложены, развиты и теоретически обоснованы новые вероятностно-информированные нейросетевые модели для анализа малых, неоднородных и несбалансированных наборов изображений с неизвестной сложной стохастической структурой.

Исследованы теоретические основы предлагаемых методов информирования, включающие в себя обоснование способа реализации информирования и определение свойств модели, позволяющих предполагать ее эффективность, как источника дополнительной информации в конкретной задаче. Так, в задаче классификации малых наборов изображений для новой вероятностной модели факторного анализатора с аддитивным и импульсным шумами были доказаны несмещенность и состоятельность оценок параметров, полученных минимизацией кросс-энтропии. Результат аналитически обосновывает то, что оценки модели не накапливают систематической ошибки, и это подтверждается экспериментально: в 45 из 72 тестируемых конфигураций прирост точности над базовыми архитектурами был получен за счет моделирования в сети аддитивного шума, а в 65 из 72 – импульсного. Реализация информирования на уровне архитектуры сети при этом напрямую следует из структуры модели и необходимости моделирования с ее помощью глобальных признаков снимка.

Для моделирующего локальные пространственные связи между пикселями изображения случайного поля Маркова, имеющего форму квадродерева, доказано свойство эргодичности при условии, что граф поля является неориентированным. Результат обосновывает преимущество использования таких сетей для анализа изображений, содержащих разномасштабные объекты – свойство эргодичности позволяет переносить выделенные закономерности на разные разрешения снимка. Из доказанной теоремы о связи между алгоритмом расчета вероятностей в модели квадродерева и графовой сетью следует выбор способа реализации информирования этой моделью – на уровне архитектуры сети с помощью слоев графовой свертки.

В задаче обработки неоднородных наборов изображений доказано, что использование в качестве дополнительных входных признаков вероятностей каждого элемента данных принадлежать компонентам нормальной смеси, моделирующей яркости пикселей снимка, уменьшает ошибку восстановления

целевой функции в сравнении со случаем сети без информирования. Из этой теоремы также следует, что информирование с помощью смеси вероятностных законов должно быть реализовано на уровне входных признаков.

В работе проведены исследования аналитических свойств информированых нейросетевых архитектур. Для классификатора, информированного моделью факторного анализатора с аддитивным и импульсным шумами было доказано, что такая сеть обладает меньшей вычислительной сложностью, в сравнении с базовыми классификаторами. Результат подтверждается экспериментально: количество параметров сети уменьшается по сравнению с базовым классификатором вплоть до 496 тысяч, а сокращение количества выполняемых операций достигает 1.343 миллиона FLOPS.

Оценена динамика обучения сетей, информированных на уровне архитектуры моделью поля Маркова в виде квадродерева. Доказано, что информированная графовая сеть способна обучаться быстрее, чем архитектура, выполняющую свертку по двумерной решетке, традиционно применяющейся для моделирования изображений. Данный результат также справедлив и для снимков высокого разрешения, когда в качестве узлов графа рассматриваются суперпиксели. Также доказано, что использование в информированной квадродеревом архитектуре дополнительной ветви, реализующей обработку локальных внутри-суперпиксельных связей, повышает скорость обучения сети в сравнении с сопоставимыми линейными и нелинейными графовыми архитектурами, в том числе использующими механизмы внимания.

Продемонстрирована высокая эффективность разработанных информированных нейросетевых моделей в прикладных задачах анализа изображений, в особенности аэрокосмических снимков земной поверхности, полученных помощью спутников и БПЛА. При этом созданные методы информирования нейронных сетей не являются узкоспециализированными, поскольку они построены на основе общих вероятностных моделей снимков.

Классификатор FtFNN, информированный на уровне архитектуры моделью факторного анализатора с аддитивным и импульсным шумами, демонстрирует более высокую точность результатов в сравнении со всеми рассмотренными базовыми сверточными нерйросетевыми архитектурами. Максимальные приросты Top-1, Top-3 и Top-5 Accuracy составляют 16.9%, 10.23% и 5.67%. Согласно тесту Фридмана, при уровне значимости 0.01 разница в значениях Top-1 Accuracy, полученных базовыми архитектурами и FtFNN, является ста-

тистически значимой, что указывает на устойчивую тенденцию к повышению точности классификации с помощью FtFNN.

Разработанная для сегментации неоднородных наборов изображений нейросетевая архитектура PrINN реализует концепцию информирования композицией моделей конечной смеси вероятностных распределений и поля Маркова в форме квадродерева. Разработанная архитектура была протестирована для сегментации реальных радиолокационных изображений, полученных радиолокаторами Sentinel-1, ESAR и Capella. Прирост точности сегментации снимков достигает 20.31% по метрике Accuracy и 19.24% по метрике F_1 в сравнении со сверточными и трансформерными архитектурами. Кроме того, PrINN сегментирует изображения с более высокой точностью, чем архитектуры, информированные смесью или квадродеревом по отдельности.

Для обработки сильно несбалансированных наборов изображений была реализована новая графово-сверточная ансамблевая архитектура FN-QiGSAN, информированная на уровне архитектуры полем Маркова в виде квадродерева. Результаты многоклассовой сегментации аэрокосмических снимков высокого разрешения показали, что FN-QiGSAN повышает точность обработки изображений в сравнении со всеми базовыми сверточными и трансформерными сегментаторами. Средний прирост значений F_1-меры для крупных объектов достигает 14.67%, а для малых (автомобили) – 11.89%. FN-QiGSAN демонстрирует более высокое качество обработки изображений в сравнении с SOTA-моделью LWGANet: для некоторых классов прирост F_1-меры достигает 15.11%. FN-QiGSAN также демонстрирует превосходящую точность выделения кораблей и автомобилей при двухклассовой сегментации. Прирост средних значений F_1 составляет 4.16-66.24%. При этом лучшие по точности конфигурации FN-QiGSAN в задаче двухклассовой сегментации содержат до 13.4 раз меньше параметров чем лучшие по точности базовые сети.

Созданы комплексы программных решений на языках программирования C++ и Python, реализующие статистические алгоритмы, методы искусственного интеллекта и работу с нейронными сетями. Они предназначены для автоматизации моделирования, проведения анализа данных и обработки значительных объемов изображений. Для обучения рассматриваемых в диссертации нейросетевых архитектур задействовались гибридные высокопроизводительные вычислительные ресурсы центра коллективного пользования «Информатика»

Федерального исследовательского центра «Информатика и управление» Российской академии наук.

Все поставленные задачи исследования были решены для разных типов ограниченных наборов данных. Все полученные результаты являются новыми, а проведенные исследования – комплексными и в достаточной степени универсальными. Они не требуют существенных априорных предположений о свойствах и закономерностях обрабатываемых изображений, связанных со спецификой предметной области. При этом разработанные методы демонстрируют высокую эффективность в условиях ограниченности информации об изучаемом объекте. Таким образом, созданные вероятностно-информированные модели могут являться основой для создания нейросетевых методов анализа научных и технических данных сложной структуры.

Список литературы

1. *Aleiassaee A.*, *Kumar A.*, *al et.* Transformers in Remote Sensing: A Survey // Remote Sensing. — 2023. — Т. 15, № 7. — С. 1860.
2. *Yang K.*, *Wu Z.*, *Arnold F.* Machine-learning-guided directed evolution for protein engineering // Nature Methods. — 2019. — Т. 16. — С. 687—694.
3. *Gliner V.* [и др.]. Using domain adaptation for classification of healthy and disease conditions from mobile-captured images of standard 12-lead electrocardiograms // Scientific Reports. — 2023. — Т. 13, № 1. — С. 14023.
4. *Визильтер Ю.*, *Желтов С.*, *Бусурин В.* Современный морфологический анализ и его применение в авиационных системах технического зрения. — Москва : Московский авиационный институт (национальный исследовательский университет), 2020. — С. 160.
5. *Горшенин А.*, *Королев В.* Аппроксимация распределений размеров частиц лунного реголита на основе метода статистической симуляции выборок // Информатика и ее применения. — 2020. — Т. 14, № 2. — С. 50—57.
6. *Belyaev K.* [и др.]. Some Features of the Intra-Annual Variability of Heat Fluxes in the North Atlantic // Izvestiya - Atmospheric and Oceanic Physics. — 2021. — Т. 57, № 6. — С. 619—631.
7. *Karpov K.*, *Korolev V.*, *Sukhareva N.* Statistical separation of mixtures in the problem of reconstructing the coefficients of an Itô stochastic process-type model of the interplanetary magnetic flux density: L2-distance minimization vs likelihood maximization // Russian Journal of Numerical Analysis and Mathematical Modelling. — 2025. — Т. 40, № 1. — С. 17—31.
8. *Pearson K.* The Grammar of Science. — Adam, Charles Black, 1900. — С. 548.
9. *Fukunaga K.* Introduction to Statistical Pattern Recognition. — New York : Academic Press, 1972. — С. 369.
10. *Колмогоров А. Н.* Теория вероятностей и математическая статистика. — Москва : Наука, 1986. — С. 534.

11. *Харкевич А. А.* Теория информации и ее приложения: Сборник переводов иностранных статей по теории информации и ее приложениям к связи. — Москва : Гос. изд. физико-математической литературы, 1959. — С. 329.

12. *Журавлев Ю. И.*, *Гуревич И. Б.* Распознавание образов и распознавание изображений // Распознавание, классификация, прогноз. Математические методы и их применение. — М. : Наука, 1989. — С. 5—72.

13. *Рудаков К.* Об алгебраической теории универсальных и локальных ограничений для задач классификации // Распознавание, классификация, прогноз. М.: Наука. — 1989. — С. 176—201.

14. *Rosenblatt F.* The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain // Psychological Review. — 1958. — Т. 65. — С. 386—408.

15. *Rumelhart D. E.*, *Hinton G. E.*, *Williams R. J.* Learning representations by back-propagating errors // Nature. — 1986. — Т. 323, № 6088. — С. 533.

16. *Lecun Y.* [и др.]. Gradient-based learning applied to document recognition // Proceedings of the IEEE. — 1998. — Т. 86, № 11. — С. 2278—2324.

17. *Hochreiter S.*, *Schmidhuber J.* Long Short-Term Memory // Neural Computation. — 1997. — Т. 9, № 8. — С. 1735—1780.

18. *Bengio Y.* [и др.]. A neural probabilistic language model // Journal of Machine Learning Research. — 2003. — Т. 3. — С. 1137—1155.

19. *Wang J.* [и др.]. VGGT: Visual Geometry Grounded Transformer // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2025.

20. *Minaee S.* [и др.]. Image Segmentation Using Deep Learning: A Survey // IEEE transactions on pattern analysis and machine intelligence. — 2021. — Т. 44, № 7. — С. 3523—3542.

21. *Chen L.* [и др.]. Review of Image Classification Algorithms Based on Convolutional Neural Networks // Remote Sensing. — 2021. — Т. 13, № 22. — С. 4712.

22. *Li J.*, *Wei X.* Research on efficient detection network method for remote sensing images based on self attention mechanism // Image and Vision Computing. — 2024. — Т. 142. — С. 104884.

23. *Lan K.* [и др.]. High-Efficiency and High-Precision Ship Detection Algorithm Based on Improved YOLOv8n // Mathematics. — 2017. — Т. 12, № 7. — С. 1072.

24. *Manocha A., Afaq Y., Bhatia M.* Mapping of water bodies from Sentinel-2 images using deep learning-based feature fusion approach // Neural Computing & Applications. — 2023. — Т. 35. — С. 9167—9179.

25. *Freudenberg M., Magdon P., Nolke N.* Individual tree crown delineation in high-resolution remote sensing images based on U-Net // Neural Computing & Applications. — 2022. — Т. 34. — С. 22197—22207.

26. *Bertels J.* [и др.]. Convolutional neural networks for medical image segmentation. — 2022. — arXiv: 2211 . 09562. — URL: https : / / arxiv . org/abs/2211.09562.

27. *Chen L.-C.* [и др.]. Rethinking Atrous Convolution for Semantic Image Segmentation // 2017 Conference on Computer Vision and Pattern Recognition (CVPR). — 2017.

28. *Dosovitskiy A.* [и др.]. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. — 2021. — arXiv: 2010.11929. — URL: https://arxiv.org/abs/2010.11929.

29. *Strudel R.* [и др.]. Segmenter: Transformer for Semantic Segmentation. — 2021. — URL: https://arxiv.org/abs/2105.05633.

30. *Chen H., Qi Z., Shi Z.* Remote Sensing Image Change Detection With Transformers // IEEE Transactions on Geoscience and Remote Sensing. — 2022. — Т. 60.

31. *Li Z.* [и др.]. MLANet: A Robust Ship Segmentation Network Based on Multilevel Multiattention Feature Fusion for Complex Maritime Background Environments // IEEE Sensors Journal. — 2024. — Т. 24, № 24. — С. 42404—42416.

32. *Shamshad F.* [и др.]. Transformers in medical imaging: A survey // Medical Image Analysis. — 2023. — Т. 88. — С. 102802.

33. *Touvron H.* [и др.]. Training data-efficient image transformers : distillation through attention // Proceedings of the 38th International Conference on Machine Learning. Т. 139 / под ред. M. Meila, T. Zhang. — 18–24 Jul.2021. — С. 10347—10357. — (Proceedings of Machine Learning Research).

34. *Zhou D.* [и др.]. Refiner: Refining Self-attention for Vision Transformers // arXiv. — 2021. — Т. abs/2106.03714. — URL: https://arxiv.org/abs/2106.03714.

35. *Roy S.* [и др.]. Multimodal Fusion Transformer for Remote Sensing Image Classification // IEEE Transactions on Geoscience and Remote Sensing. — 2023. — Т. 61.

36. *Lu W.* [и др.]. LWGANet: A Lightweight Group Attention Backbone for Remote Sensing Visual Tasks // arXiv e-prints. — 2025.

37. *Wang Y.* [и др.]. Vision Transformers for Image Classification: A Comparative Survey // Technologies. — 2025. — Т. 13, № 1.

38. *Zheng Z.* [и др.]. Graph-Transformer with spatial-spectral features fusion for hyperspectral image classification // Expert Systems with Applications. — 2025. — Т. 264. — С. 125962.

39. *He X.* [и др.]. Transformer Embedding UNet for Remote Sensing Image Semantic Segmentation // IEEE Transactions on Geoscience and Remote Sensing. — 2022. — Т. 60.

40. *Chu X.* [и др.]. Do We Really Need Explicit Position Encodings for Vision Transformers? — 02.2021. — arXiv: 2102.10882.

41. *Liu Z.* [и др.]. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. — 2021. — arXiv: 2103.14030. — URL: https://arxiv.org/abs/2103.14030.

42. *Xu P.* [и др.]. Small data machine learning in materials science // npj Computational Materials. — 2023. — Т. 9.

43. *Safonova A.* [и др.]. Ten deep learning techniques to address small data problems with remote sensing // International Journal of Applied Earth Observation and Geoinformation. — 2023. — Т. 125. — С. 103569.

44. *Izonin I.* [и др.]. A GRNN-based Approach towards Prediction from Small Datasets in Medical Application // Procedia Computer Science. — 2021. — Т. 184. — С. 242—249. — The 12th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 4th International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.

45. *Sang S.* [и др.]. Small-Object Sensitive Segmentation Using Across Feature Map Attention // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2023. — Т. 45, № 5. — С. 6289—6306.

46. *Lee W.*, *Seo K.* Downsampling for Binary Classification with a Highly Imbalanced Dataset Using Active Learning // Big Data Research. — 2022. — Т. 28. — С. 100314.

47. *Chen L.* [и др.]. Underwater object detection in noisy imbalanced datasets // Pattern Recognition. — 2024. — Т. 155. — С. 110649.

48. *Kumar G.* [и др.]. Data Harmonization for Heterogeneous Datasets: A Systematic Literature Review // Applied Sciences. — 2021. — Т. 11, № 17.

49. *Wang C.*, *Gu H.*, *Su W.* SAR Image Classification Using Contrastive Learning and Pseudo-Labels With Limited Data // IEEE Geoscience and Remote Sensing Letters. — 2022. — Т. 19. — С. 1—5.

50. *Karpatne A.* [и др.]. Predictive Learning in the Presence of Heterogeneity and Limited // Proceedings of the 2014 SIAM International Conference on Data Mining (SDM). — С. 253—261.

51. *Турдаков Д.* [и др.]. Доверенный искусственный интеллект: вызовы и перспективные решения // Доклады Российской Академии Наук. Математика, информатика, процессы управления. — 2022. — Т. 508, № 1. — С. 13—18.

52. *Chawla N. V.* [и др.]. SMOTE: synthetic minority over-sampling technique // Journal of artificial intelligence research. — 2002. — Т. 16. — С. 321—357.

53. *Gong Z.* [и др.]. MSAug: Multi-Strategy Augmentation for rare classes in semantic segmentation of remote sensing images // Displays. — 2024. — Т. 84. — С. 102779.

54. *Che Q.-H.* [и др.]. Enhanced Generative Data Augmentation for Semantic Segmentation via Stronger Guidance. // Proceedings of the 14th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2025). — 2025. — С. 251—262.

55. *Su X.* [и др.]. A Unified GAN Framework Regarding Manifold Alignment for Remote Sensing Images Generation. — 2023. — arXiv: 2305.19507. — URL: https://arxiv.org/abs/2305.19507.

56. *Brigato L.*, *Iocchi L.* A Close Look at Deep Learning with Small Data // 2020 25th International Conference on Pattern Recognition (ICPR). — 2021. — С. 2490—2497.

57. *Genc A.*, *Kovarik L.*, *Fraser H. L.* A deep learning approach for semantic segmentation of unbalanced data in electron tomography of catalytic materials // Scientific Reports. — 2022. — Т. 12, № 1. — С. 16267.

58. *Saeedizadeh N.* [и др.]. A new optimization approach based on neural architecture search to enhance deep U-Net for efficient road segmentation // Knowledge-Based Systems. — 2024. — Т. 296. — С. 111966.

59. *Qiu S.* [и др.]. Subclassified Loss: Rethinking Data Imbalance From Subclass Perspective for Semantic Segmentation // IEEE Transactions on Intelligent Vehicles. — 2024. — Т. 9, № 1. — С. 1547—1558.

60. *Debnath R.*, *Das K.*, *Bhowmik M. K.* GSNet: A new small object attention based deep classifier for presence of gun in complex scenes // Neurocomputing. — 2025. — Т. 635. — С. 129855.

61. *Liu W.* [и др.]. SOSNet: Real-Time Small Object Segmentation via Hierarchical Decoding and Example Mining // IEEE Transactions on Neural Networks and Learning Systems. — 2025. — Т. 36, № 2. — С. 3071—3083.

62. *Kiobya T.*, *Zhou J.*, *Maiseli B.* A multi-scale semantically enriched feature pyramid network with enhanced focal loss for small-object detection // Knowledge-Based Systems. — 2025. — Т. 310. — С. 113003.

63. *Wang W.* [и др.]. Explorations of Contrastive Learning in the Field of Small Sample SAR ATR // Procedia Computer Science. — 2022. — Т. 208. — С. 190—195. — 7th International Conference on Intelligent, Interactive Systems and Applications.

64. *Yang R.* [и др.]. Learning Relation by Graph Neural Network for SAR Image Few-Shot Learning // Proceedings of the IGARSS 2020. — Waikoloa, HI, USA, 2020. — С. 1743—1746.

65. *Zhang S.* [и др.]. Graph Convolutional Networks: a Comprehensive Review // Comput. Soc. Netw. — 2019. — Т. 6, № 11. — С. 94—104.

66. *Huang Z.*, *Pan Z.*, *Lei B.* Transfer Learning with Deep Convolutional Neural Network for SAR Target Classification with Limited Labeled Data // Remote Sensing. — 2017. — Т. 9, № 9. — С. 907.

67. *Park Y.*, *Hauschild A.*, *Heider D.* Transfer learning compensates limited data, batch effects and technological heterogeneity in single-cell sequencing // NAR Genom Bioinform. — 2021. — Т. 3, № 4. — lqab104.

68. *Tai Y.* [и др.]. Few-Shot Transfer Learning for SAR Image Classification Without Extra SAR Samples // IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. — 2022. — Т. 15. — С. 2240—2253.

69. *Chen Y.* [и др.]. Cross-level interaction fusion network-based RGB-T semantic segmentation for distant targets // Pattern Recognition. — 2025. — Т. 161. — С. 111218.

70. *Garau-Luis J. J.* [и др.]. Multi-modal Transfer Learning between Biological Foundation Models // The Thirty-eighth Annual Conference on Neural Information Processing Systems. — 2024. — URL: https://openreview.net/forum?id=xImeJtdUiw.

71. *Dissanayake M. W. M. G.*, *Phan-Thien N.* Neural network-based approximations for solving partial differential equations // Communications in Numerical Methods in Engineering. — 1994. — Т. 10, № 3. — С. 195—201.

72. *Lagaris I.*, *Likas A.*, *Fotiadis D.* Artificial Neural Networks for Solving Ordinary and Partial Differential Equations // IEEE Transactions on Neural Networks. — 1998. — Окт. — С. 987—1000.

73. *Karniadakis G.* [и др.]. Physics-Informed Machine Learning // Nature Reviews Physics. — 2021. — Т. 3. — С. 422—440.

74. *Kilian P.* [и др.]. A Deep Learning Factor Analysis Model Based on Importance-Weighted Variational Inference and Normalizing Flow Priors: Evaluation within a Set of Multidimensional Performance Assessments in Youth Elite Soccer Players. // Statistical Analysis and Data Mining: The ASA Data Science Journal. — 2022. — Т. 16, № 11. — С. 474—487.

75. *Anderson T. W.*, *Rubin H.* Statistical inference in factor analysis // In Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability. Т. 15. — 1956. — С. 111—150.

76. *Liu X.*, *Zhu M.*, *Lu L. e. a.* Multi-resolution partial differential equations preserved learning framework for spatiotemporal dynamics // Communications Physics. — 2024. — Т. 7.

77. *Gao H.*, *Kaltenbach S.*, *Koumoutsakos P.* Generative learning for forecasting the dynamics of high-dimensional complex systems // Nature Communications. — 2024. — Т. 15.

78. *Raissi M.*, *Perdikaris P.*, *Karniadakis G.* Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations // Journal of Computational Physics. — 2019. — Т. 378. — С. 686—707.

79. *Vinuesa R.*, *Brunton S.* Enhancing computational fluid dynamics with machine learning // Nature Computational Sciences. — 2022. — Т. 2. — С. 358—366.

80. *Li M.*, *McComb C.* Using physics-informed generative adversarial networks to perform super-resolution for multiphase fluid simulations // Journal of Computing and Information Science in Engineering. — 2022. — Т. 22, № 4. — С. 044501.

81. *Lee J.*, *Shin S.*, *Kim T. e. a.* Physics informed neural networks for fluid flow analysis with repetitive parameter initialization // Scientific Reports. — 2025. — Т. 15.

82. *Ren Z.* [и др.]. Physics-Informed Neural Networks: A Review of Methodological Evolution, Theoretical Foundations, and Interdisciplinary Frontiers Toward Next-Generation Scientific Computing // Applied Sciences. — 2025. — Т. 15, № 14.

83. *Zhang X.*, *Tu C.*, *Yan Y.* Physics-informed neural network simulation of conjugate heat transfer in manifold microchannel heat sinks for high-power IGBT cooling // International Communications in Heat and Mass Transfer. — 2024. — Т. 159.

84. *Jagtap A. D.* [и др.]. Physics-informed neural networks for inverse problems in supersonic flows // Journal of Computational Physics. — 2022. — Т. 466. — С. 111402.

85. *Li Y.* [и др.]. A time–frequency physics-informed model for real-time motion prediction of semi-submersibles // Ocean Engineering. — 2024. — Т. 299. — С. 117379.

86. *Shi Y.*, *Beer M.* Physics-informed neural network classification framework for reliability analysis // Expert Systems with Applications. — 2024. — Т. 258. — С. 125207.

87. *Amiri-Hezaveh A.*, *Tan S.*, *Deng Q. e. a.* A Physics-Informed Deep Learning Deformable Medical Image Registration Method Based on Neural ODEs // International Journal of Computer Vision. — 2025.

88. *Belomestny D.* [и др.]. Simultaneous approximation of a smooth function and its derivatives by deep neural networks with piecewise-polynomial activations // Neural Networks. — 2023. — Т. 161. — С. 242—253.

89. *Belomestny D.*, *Naumov A.*, *Samsonov S.* Statistical analysis of Inverse Entropy-regularized Reinforcement Learning. — 2025. — arXiv: 2512.06956.

90. *Li B.*, *Zhou S.*, *Ma Q. e. a.* Physics-Informed Neural Network Based Digital Image Correlation Method // Experimental Mechanics. — 2025. — Т. 65. — С. 221—240.

91. *Huang Z.* [и др.]. Physically explainable CNN for SAR image classification // ISPRS Journal of Photogrammetry and Remote Sensing. — 2022. — Т. 190. — С. 25—37.

92. *Huang Z.*, *Yao X.*, *Han J.* Progress and Perspective on Physically Explainable Deep Learning for Synthetic Aperture Radar Image Interpretation // Journal of Radars. — 2022. — Т. 13. — С. 107—125.

93. *Zheng T.*, *Wang J.*, *Lei P.* Deep Learning Based Target Detection Method with Multi-Features in SAR Imagery // Proceedings of the 2019 6th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR). — Xiamen, China, 2019. — С. 1—4.

94. *Belloni C.* [и др.]. Pose-Informed Deep Learning Method for SAR ATR // IET Radar Sonar Navig. — 2020. — Т. 14. — С. 1649—1658.

95. *Zhang L.*, *al. et.* Domain Knowledge Powered Two-Stream Deep Network for Few-Shot SAR Vehicle Recognition // IEEE Transactions on Geoscience and Remote Sensing. — 2022. — Т. 60. — С. 1—15.

96. *Dwivedi V.*, *Srinivasan B.*, *Krishnamurthi G.* Physics informed contour selection for rapid image segmentation // Scientific Reports. — 2024. — Т. 14. — С. 6996.

97. *Vu P.-A.* [и др.]. Probabilistic and Physics-Informed Machine Learning for Predictive Maintenance with Time Series Data // 24th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE). — Graz, Austria, 2023. — С. 1—8.

98. *Tyralis H.*, *Papacharalampous G.* A review of predictive uncertainty estimation with machine learning // Artificial Intelligence Review. — 2024. — Т. 57, № 4. — С. 94.

99. *Wang Z.*, *Nakahira Y.* A Generalizable Physics-informed Learning Framework for Risk Probability Estimation // Proceedings of The 5th Annual Learning for Dynamics and Control Conference. Т. 211 / под ред. N. Matni, M. Morari, G. J. Pappas. — PMLR, 15–16 Jun.2023. — С. 358—370.

100. *Zhang Z.*, *Li J.*, *Liu B.* Annealed adaptive importance sampling method in PINNs for solving high dimensional partial differential equations // Journal of Computational Physics. — 2025. — Т. 521. — С. 113561.

101. *Zuo L.* [и др.]. A spiking neural network with probability information transmission // Neurocomputing. — 2020. — Т. 408. — С. 1—12.

102. *Kelshaw D.*, *Rigas G.*, *Magri L.* Physics-Informed CNNs for Super-Resolution of Sparse Observations on Dynamical Systems // NeurIPS Workshop on Machine Learning for the Physical Sciences (2022). — 2022.

103. *Subramanian A.*, *Mahadevan S.* Probabilistic Physics-Informed Machine Learning for Dynamic Systems // Reliability Engineering and System Safety. — 2023. — Т. 230.

104. *Fuhg J.*, *Bouklas N.* On Physics-Informed Data-Driven Isotropic and Anisotropic Constitutive Models Through Probabilistic Machine Learning and Space-Filling Sampling // Computer Methods in Applied Mechanics and Engineering. — 2022. — Т. 394. — С. 114915.

105. *Zhou T.* [и др.]. A Physically Consistent Framework for Fatigue Life Prediction Using Probabilistic Physics-Informed Neural Network // International Journal of Fatigue. — 2023. — Т. 166. — С. 107234.

106. *Markham A.* [и др.]. Neuro-Causal Factor Analysis // ICML 2023 Workshop on Structured Probabilistic Inference and Generative Modeling. — 2023. — arXiv: 2305.19802.

107. *Quadri S.*, *Sidek O.* Multisensor Data Fusion Algorithm using Factor Analysis Method // International Journal of Advanced Science and Technology. — 2013. — № 55. — С. 43—52.

108. *Lin W.* [и др.]. Self-Supervised Neural Factor Analysis for Disentangling Utterance-Level Speech Sepresentations. // Proceedings of the 40 th International Conference on Machine Learning. — Honolulu, Hawaii, USA, 2023. — С. 21065—21077.

109. *Ahfock D.*, *Pyne S.*, *McLachlan G.* Data fusion using factor analysis AND low-rank matrix completion // Statistics and Computing. — 2021. — Т. 31.

110. *Li Y.* [и др.]. Probabilistic gear fatigue life prediction based on physics-informed transformer // Expert Systems with Applications. — 2024. — Т. 249. — С. 123882.

111. *Feng F.* [и др.]. Probabilistic fatigue life prediction in additive manufacturing materials with a physics-informed neural network framework // Expert Systems with Applications. — 2025. — Т. 275. — С. 127098.

112. *Gorshenin A.*, *Kuzmin V.* Method for Improving Accuracy of Neural Network Forecasts Based on Probability Mixture Models and its Implementation as a Digital Service // Informatika i ee primeneniya. — 2021. — Т. 15, № 3. — С. 63—74.

113. *Gorshenin A.*, *Kuzmin V.* Statistical Feature Construction for Forecasting Accuracy Increase and Its Applications in Neural Network Based Analysis // Mathematics. — 2022. — Т. 10. — С. 589.

114. *Gorshenin A.*, *Vilyaev A.* Finite Normal Mixture Models for the Ensemble Learning of Recurrent Neural Networks with Applications to Currency Pairs // Pattern Recognit. Image Anal. — 2022. — Т. 32. — С. 780—792.

115. *Gorshenin A.*, *Vilyaev A.* Machine Learning Models Informed by Connected Mixture Components for Short- and Medium-Term Time Series Forecasting // AI. — 2024. — Т. 5, № 4. — С. 1955—1976.

116. *Gorshenin A.* [и др.]. Mobile network traffic analysis based on probability-informed machine learning approach // Computer Networks. — 2024. — Т. 247. — С. 110433.

117. *Viroli C.*, *Mclachlan G. J.* Deep Gaussian mixture models // Statistics and Computing. — 2019. — Т. 29, № 1. — С. 43—51.

118. *Batanov G.* [и др.]. The evolution of probability characteristics of low-frequency plasma turbulence // Mathematical Models and Computer Simulations. — 2012. — Т. 4, № 1. — С. 10—25.

119. *Batanov G.* [и др.]. Evolution of statistical properties of microturbulence during transient process under electron cyclotron resonance heating of the L-2M stellarator plasma // Plasma Physics and Controlled Fusion. — 2019. — Т. 61, № 7. — С. 075006.

120. *Gnip P Vokorokos L D. P.* Selective oversampling approach for strongly imbalanced data // PeerJ Computer Science. — 2021. — Т. 7. — e604.

121. *Zhou Y.* [и др.]. SAR Target Classification with Limited Data via Data Driven Active Learning // IGARSS 2020–2020 IEEE International Geoscience and Remote Sensing Symposium. — 2020. — С. 2475—2478.

122. *Dostovalova A.* Using a Model of a Spatial-Hierarchical Quadtree with Truncated Branches to Improve the Accuracy of Image Classification // Izvestiya, Atmospheric and Oceanic Physics. — 2023. — Т. 59, № 12. — С. 1—8.

123. *Достовалова А.* Нейросетевое квадродерево и его применение для сегментирования спутниковых изображений // Информатика и ее применения. — 2024. — Т. 18, № 4. — С. 77—85.

124. *Dostovalova A., Gorshenin A.* Neural Network Image Classifiers Informed by Factor Analyzers // Doklady Mathematics. — 2024. — Т. 110, Suppl.1. — S35—S41.

125. *Dostovalova A., Gorshenin A.* Small sample learning based on probability-informed neural networks for SAR image segmentation // Neural Computing and Applications. — 2025. — Т. 37. — С. 8285—8308.

126. *Горшенин А., Достовалова А.* MMRFiGN: ансамблевая графовая модель сегментации несбалансированных изображений высокого разрешения, информированная мультикомпонентными Марковскими случайными полями // Доклады Российской академии наук. Математика, информатика, процессы управления. — 2025. — Т. 527, № 9. — С. 156—170.

127. *Gorshenin A., Dostovalova A.* QiGSAN: A Novel Probability-Informed Approach for Small Object Segmentation in the Case of Limited Image Datasets // Big Data and Cognitive Computing. — 2025. — Т. 9, № 9.

128. *Достовалова А.*, *Горшенин А.* О сегментации малых объектов на радиолокационных изображениях при помощи графово-сверточных сетей, информированных квадродеревом // Современные проблемы дистанционного зондирования Земли. — 2025. — Т. 1, № 1. — С. 11.

129. *Достовалова А.* [и др.]. Сравнительный анализ модификаций нейросетевых архитектур U-Net в задаче сегментации медицинских изображений // Digital diagnostics. — 2024. — Т. 5, № 4. — С. 833—853.

130. *Dostovalova A.*, *Gorshenin A.* QuadTree-Based Graph Convolutional Networks for Small Object Segmentation // Communications in Computer and Information Science. — 2025.

131. *Горшенин А.*, *Достовалова А.* О композиции графово-сверточных нейронных сетей и квадродеревьев в задаче сегментации кораблей на радиолокационных изображениях // Ломоносовские чтения. — 2025. — С. 93—94.

132. *Горшенин А.*, *Достовалова А.* Нейросетевые аналоги случайных полей маркова в задачах сегментации объектов на спутниковых снимках // Тихоновские чтения. — 2024. — С. 114.

133. *Достовалова А.* О применении глубоких гауссовских смешанных моделей в задачах классификации и регрессии временных рядов и табличных данных // Материалы Международного молодежного научного форума "ЛОМОНОСОВ-2025". Секция "Вычислительная математика и кибернетика". — 2025.

134. *Достовалова А.* Совместное применение нейронных сетей и вероятностных моделей для сегментирования радиолокационных изображений // Материалы Международного молодежного научного форума "ЛОМОНОСОВ-2024". Секция "Вычислительная математика и кибернетика". — 2024.

135. *Tan M.*, *Le Q. V.* EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks // Proceedings of the 36th International Conference on Machine Learning, ICML 2019. — Long Beach, 9-15 June, 2019. — С. 6105—61148.

136. *Chollet F.* Xception: Deep Learning with Depthwise Separable Convolutions // 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2016. — С. 1800—1807.

137. *McLachlan G.*, *Lee S.*, *Rathnayake S.* Finite Mixture Models // Annual Review of Statistics and Its Application. — 2019. — Т. 6. — С. 355—378.

138. *Kato Z.*, *Zerubia J.* Markov Random Fields in Image Segmentation // Foundations and Trends in Signal Processing. — 2011. — Т. 5. — С. 1—155.

139. *Paszke A.* [и др.]. Enet: A deep neural network architecture for real-time semantic segmentation. — 2016. — arXiv: 1606.02147.

140. *Zhao H.* [и др.]. Pyramid Scene Parsing Network // 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2017. — С. 6230—6239.

141. *Spasev V.* [и др.]. Semantic Segmentation of Unmanned Aerial Vehicle Remote Sensing Images Using SegFormer // Intelligent Systems and Pattern Recognition. Communications in Computer and Information Science. Т. 2305. — 2024. — С. 1416—1425.

142. *Lee J.* [и др.]. Deep learning for rare disease: A scoping review // Journal of Biomedical Informatics. — 2022. — Т. 135. — С. 104227.

143. *Piffer S.* [и др.]. Tackling the small data problem in medical image classification with artificial intelligence: a systematic review // Progress in Biomedical Engineering. — 2024. — Т. 6, № 3. — С. 032001.

144. *Misnik A.*, *Kapelko E.* Low-Data Welding Defects Detection // 2024 7th International Conference on Information Technologies in Engineering Education (Inforino). — 2024. — С. 1—6.

145. *Ayana G.* [и др.]. A Novel Multistage Transfer Learning for Ultrasound Breast Cancer Image Classification // Diagnostics. — 2022. — Т. 12, № 1.

146. *Bilic P.* [и др.]. The Liver Tumor Segmentation Benchmark (LiTS) // Medical Image Analysis. — 2023. — Т. 84.

147. *Huang L.* [и др.]. Multi-Scale Feature Fusion Convolutional Neural Network for Indoor Small Target Detection // Frontiers in Neurorobotics. — 2022. — Т. 16.

148. *Hsieh S.*, *Cheng Y.* Multimodal feature fusion in deep learning for comprehensive dental condition classification // J Xray Sci Technol. — 2024. — Т. 32, № 2. — С. 303—321.

149. *Liu J.* [и др.]. Deep feature fusion classification network (DFFCNet): Towards accurate diagnosis of COVID-19 using chest X-rays images // Biomed Signal Process Control. — 2022. — Т. 76. — С. 103677.

150. *Kwak Y.* [и др.]. Multilevel Feature Fusion With 3D Convolutional Neural Network for EEG-Based Workload Estimation // IEEE Access. — 2020. — Т. 8. — С. 16009—16021.

151. *Wang D.* [и др.]. A novel deep-learning based weighted feature fusion architecture for precise classification of pressure injury // Frontiers in Physiology. — 2024. — Т. 15.

152. *Reinders C.*, *Schubert F.*, *Rosenhahn B.* HydraMix: Multi-Image Feature Mixing for Small Data Image Classification. — 01.2025. — arXiv: 2501.09504.

153. *Verbeek J.* Learning nonlinear image manifolds by global alignment of local linear models // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2006. — Т. 28, № 8. — С. 1236—1250.

154. *Legin R.*, *Adam A.*, *Perreault-Levasseur Y. H. L.* Beyond Gaussian Noise: A Generalized Approach to Likelihood Analysis with Non-Gaussian Noise // The Astrophysical Journal Letters. — 2023. — Т. 949, № 2.

155. *Tang Y.*, *Salakhutdinov R.*, *Hinton G.* Deep Mixtures of Factor Analysers // Proceedings of the 29th International Conference on Machine Learning. — Edinburgh, Scotland, UK, 2012.

156. *Lee S.*, *Lin T.*, *McLachlan G.* Mixtures of factor analyzers with scale mixtures of fundamental skew normal distributions // Adv Data Anal Classif. — 2021. — Т. 15. — С. 481—512.

157. *Hu H.*, *Li B.*, *Liu Q.* Removing Mixture of Gaussian and Impulse Noise by PatchBased Weighted Means // Journal of Scientific Computing. — 2016. — № 67. — С. 103—129.

158. *Kusnik D.*, *Smolka B.* Robust mean shift filter for mixed Gaussian and impulsive noise reduction in color digital images // Nature Scientic Reports. — 2022. — № 12. — С. 14951.

159. *Shongwe T.*, *Vinck A. J. H.*, *Ferreira H. C.* A Study on Impulse Noise and Its Models // SAIEE Africa Research Journal. — 2015. — Т. 106, № 3. — С. 119—131.

160. *Amrouche M.*, *Carfantan H.*, *Idier J.* Efficient Sampling of Bernoulli-Gaussian-Mixtures for Sparse Signal Restoration // IEEE Transactions on Signal Processing. — 2022. — Т. 70. — С. 5578—5591.

161. *Yakowitz S.*, *Spragins J.* On the identifiability of finite mixtures // Annals of Mathematical Statistics. — 1968. — Т. 39. — С. 209—214.

162. *Fukshansky L.* On Effective Witt Decomposition and the Cartan–Dieudonne Theorem // Canadian Journal of Mathematics. — 2007. — Т. 59, № 6. — С. 1284—1300.

163. *Eguchi S.*, *Copas J.* Interpreting Kullback–Leibler divergence with the Neyman–Pearson lemma // Journal of Multivariate Analysis. — 2006. — Т. 97, № 9. — С. 2034—2040. — Special Issue dedicated to Prof. Fujikoshi.

164. *Gimpel D. H. A. K.* Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units. — 2016. — arXiv: 1606.08415. — URL: http://arxiv.org/abs/1606.08415.

165. *Shridhar K.* [и др.]. ProbAct: A Probabilistic Activation Function for Deep Neural Networks. — 2020. — arXiv: 1905.10761.

166. *Howard A.* [и др.]. Searching for MobileNetV3 // 2019 IEEE/CVF International Conference on Computer Vision (ICCV). — 2019. — С. 1314—1324.

167. *Howard A. G.* [и др.]. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. — 2017. — arXiv: 1704.04861.

168. *Sandler M.* [и др.]. MobileNetV2: Inverted Residuals and Linear Bottlenecks // The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2018. — С. 4510—4520.

169. *He K.* [и др.]. Deep Residual Learning for Image Recognition // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2016. — С. 770—778.

170. *Petersen F.* [и др.]. Differentiable Top-k Classification Learning // Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA. Т. 162. — 2022.

171. *Deng J.* [и др.]. ImageNet: A large-scale hierarchical image database // 2009 IEEE Conference on Computer Vision and Pattern Recognition. — 2009. — С. 248—255.

172. *Nilsback M.-E., Zisserman A.* Automated Flower Classification over a Large Number of Classes // 2008 Sixth Indian Conference on Computer Vision, Graphics and Image Processing. — 2008. — С. 722—729.

173. *Parkhi O. M.* [и др.]. Cats and dogs // 2012 IEEE Conference on Computer Vision and Pattern Recognition. — 2012. — С. 3498—3505.

174. *Howard J., Gugger S.* Fastai: A Layered API for Deep Learning // Information. — 2020. — Т. 11, № 2.

175. *Yang Y., Newsam S.* Bag-of-visual-words and spatial extensions for land-use classification // ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS). — 11.2010. — С. 270—279.

176. *Krizhevsky A., Hinton G.* Learning multiple layers of features from tiny images. — Toronto, Ontario, 2009.

177. *Hutter I. L. A. F.* Fixing Weight Decay Regularization in Adam. — 2017. — arXiv: 1711.05101. — URL: http://arxiv.org/abs/1711.05101.

178. *Ruder S.* An overview of gradient descent optimization algorithms. — 2016. — arXiv: 1609.04747. — URL: http://arxiv.org/abs/1609.04747.

179. *M. F.* A Comparison of Alternative Tests of Significance for the Problem of M Rankings // The Annals of Mathematical Statistics. — 1940. — Т. 11, № 1. — С. 86—92.

180. *Wang X.* [и др.]. Few-Shot SAR Ship Image Detection Using Two-Stage Cross-Domain Transfer Learning // IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium. — 2022. — С. 2195—2198.

181. *Li H., Wang T., Wang S.* Few-Shot SAR Target Classification Combining Both Spatial and Frequency Information // GLOBECOM 2022 - 2022 IEEE Global Communications Conference. — 2022. — С. 480—485.

182. *Yang R.* [и др.]. Learning Relation by Graph Neural Network for SAR Image Few-Shot Learning // IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium. — 2020. — С. 1743—1746.

183. *Santoso H., Nakamura K.* Discrimination of Sidewalk Surface Condition Based on Image Textures and Meteorological Information // Journal of Advanced Computational Intelligence and Intelligent Informatics. — 2007. — Июнь. — Т. 11. — С. 491—501.

184. *Descombes X.* Chapter 11 - Markov Models and MCMC Algorithms in Image Processing // Academic Press Library in Signal Processing: Volume 4. Т. 4 / под ред. J. Trussell [и др.]. — Elsevier, 2014. — С. 293—325. — (Academic Press Library in Signal Processing).

185. *Gao G.* Statistical Modeling of SAR Images: A Survey // Sensors. — 2010. — Т. 10, № 1. — С. 775—795.

186. *Dempster A., Laird N., Rubin D.* Maximum Liklihood from Incomplete Data via the EM Algorithm // Journal of the Royal Statistical Society. Series B. — 1977. — Т. 39. — С. 1—38.

187. *Celeux G., Diebolt J.* The SEM Algorithm: a Probabilistic Teacher Algorithm Derived from the EM Algorithm for the Mixture Problem // Computational statistics quarterly. — 1985. — Т. 2, № 1. — С. 73—82.

188. *Gorshenin A., Korolev V., Tursunbayev A.* Median Modifications of the EM-Algorithm for Separation of Mixtures of Probability Distributions and Their Applications to the Decomposition of Volatility of Financial Indexes // Journal of Mathematical Sciences. — 2017. — Т. 2. — С. 176—195.

189. *Dostovalova A.* Simulation of Locally Homogeneous Radar Images Using Different Statistical Criteria // Matematicheskoe modelirovanie i chislennye metody. — 2021. — Т. 4. — С. 103—120.

190. *Королев В. Ю.* Вероятностно-статистические методы декомпозиции волатильности хаотических процессов. — Москва : Издательство Московского университета, 2011. — С. 512.

191. *Gorshenin A.* [и др.]. Coordinate-Wise versions of the Grid Method for the Analysis of Intensities of Non-Stationary Information Flows by Moving Separation of Mixtures of Gamma-Distribution // Proceedings of 27^{th} European Conference on Modelling and Simulation. — Alesund, Norway, 2013. — С. 565—568.

192. *Marron J.*, *Wand M.* Exact Mean Integrated Squared Error // The Annals of Statistics. — 1992. — Т. 20, № 2. — С. 712—736.

193. *Pastorino M.* [и др.]. Multisensor and Multiresolution Remote Sensing Image Classification through a Causal Hierarchical Markov Framework and Decision Tree Ensembles // Remote Sensing. — 2021. — Т. 13, № 5. — С. 849.

194. *Pastorino M.* [и др.]. Semantic Segmentation of Remote-Sensing Images Through Fully Convolutional Neural Networks and Hierarchical Probabilistic Graphical Models // IEEE Transactions on Geoscience and Remote Sensing. — 2022. — Т. 60. — С. 1—16.

195. *Pastorino M.* [и др.]. Classification of Multimission SAR Images Based on Probabilistic Graphical Models and Convolutional Neural Networks // Proceedings of 2023 IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2023). — 2023. — С. 1420—1423.

196. *Ronneberger O.*, *Fischer P.*, *Brox T.* U-net: Convolutional networks for biomedical image segmentation // Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. — Springer. 2015. — С. 234—241.

197. *Дуб Д. Л.* Вероятностные процессы. — Москва : Издательство иностранной литературы, 1956. — С. 605.

198. *Jetley S.* [и др.]. Learn To Pay Attention. — 2018. — arXiv: 1804.0239. — URL: https://arxiv.org/abs/1804.02391.

199. *Potin P.* [и др.]. Sentinel-1 Mission Operations Concept // Proceedings of IEEE International Geoscience and Remote Sensing Symposium. — Munich, Germany, 2012. — С. 1745—17489.

200. *Castelletti D.* [и др.]. Capella Space VHR SAR Constellation: Advanced Tasking Patterns and Future Capabilities // Proceedings of 2022 IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2022). — Kuala Lumpur, Malaysia, 2022. — С. 4137—4140.

201. *Scheiber R.* [и др.]. Recent Developments and Applications of Multi-Pass Airborne Interferometric SAR Using the E-SAR System // Proceedings of 7th European Conference on Synthetic Aperture Rada. — Friedrichshafen, Germany, 2008. — С. 1—4.

202. *Wei S.* [и др.]. HRSID: A High-Resolution SAR Images Dataset for Ship Detection and Instance Segmentation // IEEE Access. — 2020. — Т. 8. — С. 120234—120254.

203. *Bergstra J., Bengio Y.* Random Search for Hyper-Parameter Optimization // Journal of Machine Learning Research. — 2012. — Т. 13, № 10. — С. 281—305.

204. *Chitta K., Álvarez J. M., Hebert M.* Quadtree Generating Networks: Efficient Hierarchical Scene Parsing with Sparse Convolutions // 2020 IEEE Winter Conference on Applications of Computer Vision (WACV). — 2020. — С. 2009—2018.

205. *Jayaraman P. K.* [и др.]. Quadtree Convolutional Neural Networks // Computer Vision – ECCV 2018 / под ред. V. Ferrari [и др.]. — Cham : Springer International Publishing, 2018. — С. 554—569.

206. *Ronen T., Levy O., Golbert A.* Vision Transformers with Mixed-Resolution Tokenization // 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). — 2023. — С. 4613—4622.

207. *Tang S.* [и др.]. Quadtree Attention for Vision Transformers // International Conference on Learning Representations. — 2022. — URL: https://openreview.net/forum?id=fR-EnKWL%5C_Zb.

208. *Zhou J.* [и др.]. Graph neural networks: A review of methods and applications // AI Open. — 2020. — Т. 1. — С. 57—81.

209. *Jia Z.* [и др.]. Recent Research Progress of Graph Neural Networks in Computer Vision // Electronics. — 2025. — Т. 14, № 9.

210. *Zi W.* [и др.]. SGA-Net: Self-Constructing Graph Attention Neural Network for Semantic Segmentation of Remote Sensing Images // Remote Sensing. — 2021. — Т. 13, № 21. — С. 4201.

211. *Cao P.* [и др.]. DIGCN: A Dynamic Interaction Graph Convolutional Network Based on Learnable Proposals for Object Detection // Journal of artificial intelligence research. — 2024. — Т. 79. — С. 1091—1112.

212. *Csillik O.* Fast Segmentation and Classification of Very High Resolution Remote Sensing Data Using SLIC Superpixels // Remote Sensing. — 2017. — Т. 9, № 3.

213. *Xu K.* [и др.]. Optimization of Graph Neural Networks: Implicit Acceleration by Skip Connections and More Depth // Proceedings of the 38th International Conference on Machine Learning, PMLR. Т. 139. — 2021.

214. *Deift P.*, *Its A.*, *Krasovsky I.* Toeplitz matrices and toeplitz determinants under the impetus of the ising model: Some history and some recent results // Communications on Pure and Applied Mathematicss. — 2013. — Т. 66, № 9. — С. 1360—1438.

215. *S. C.* [и др.]. Analysis of Image Quality using Sobel Filter // 019 Third International Conference on Inventive Systems and Control (ICISC). — 2019. — С. 526—531.

216. *Yang F.*, *Ma Z.*, *Xie M.* Image classification with superpixels and feature fusion method // Journal of Electronic Science and Technology. — 2021. — Т. 19, № 1.

217. *Liu X.* [и др.]. Dark Spot Detection from SAR Images Based on Superpixel Deeper Graph Convolutional Network // Remote Sensing. — 2022. — Т. 14, № 21. — С. 5618.

218. *Peng Y.* [и др.]. Unifying topological structure and self-attention mechanism for node classification in directed networks // Scientific Reports. — 2025. — Т. 15, № 1. — С. 805.

219. *Wei S.* [и др.]. HRSID: A High-Resolution SAR Images Dataset for Ship Detection and Instance Segmentation // IEEE Access. — 2020. — Т. 8. — С. 120234—120254.

220. *Zhang T.* [и др.]. SAR Ship Detection Dataset (SSDD): Official Release and Comprehensive Data Analysis // Remote Sensing. — 2021. — Т. 13, № 18. — С. 3690.

221. *Lyu Y.* [и др.]. UAVid: A semantic segmentation dataset for UAV imagery // ISPRS Journal of Photogrammetry and Remote Sensing. — 2020. — Т. 165. — С. 108—119.

222. *Chen Y.* [и др.]. Large-scale structure from motion with semantic constraints of aerial images // Chinese Conference on Pattern Recognition and Computer Vision (PRCV). — Springer. 2018. — С. 347—359.

223. *Qin F.* Blind Single-Image Super Resolution Reconstruction with Gaussian Blur // Mechatronics and Automatic Control Systems / под ред. W. Wang. — Cham : Springer International Publishing, 2014. — С. 293—301.

224. *Sudre C.* [и др.]. Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations. — 2017. — Июль. — arXiv: 1707.03237.

225. *Long J.*, *Shelhamer E.*, *Darrell T.* Fully convolutional networks for semantic segmentation // 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2015. — С. 3431—3440.

226. *Sang S.* [и др.]. Small-Object Sensitive Segmentation Using Across Feature Map Attention // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2023. — Т. 45, № 5. — С. 6289—6306.

227. *Tanis J. H.*, *Giannella C.*, *Mariano A. V.* Introduction to Graph Neural Networks: A Starting Point for Machine Learning Engineers. — 2024. — arXiv: 2412.19419.

228. *Vrahatis A. G.*, *Lazaros K.*, *Kotsiantis S.* Graph Attention Networks: A Comprehensive Review of Methods and Applications // Future Internet. — 2024. — Т. 16, № 9.

229. *Hamilton W.*, *Ying Z.*, *Leskovec J.* Inductive representation learning on large graphs // Advances in neural information processing systems. — 2017. — Т. 30.

230. *Jiang M.* [и др.]. Self-attention empowered graph convolutional network for structure learning and node embedding // Pattern Recognition. — 2024. — Т. 153. — С. 110537.

231. *Ihalage A.*, *Hao Y.* Formula Graph Self-Attention Network for Representation-Domain Independent Materials Discovery // Advanced Science. — 2022. — Т. 9, № 18. — С. 2200164.