

Федеральное государственное учреждение Федеральный
исследовательский центр «Информатика и управление»
Российской академии наук (ФИЦ ИУ РАН)

На правах рукописи

Шешкус Александр Владимирович

**Использование преобразования Хафа в качестве слоя
нейронной сети**

Специальность 1.2.2 —
«Математическое моделирование, численные методы и комплексы
программ»

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
кандидат физико-математических наук
Николаев Дмитрий Петрович

Москва — 2023

Оглавление

	Стр.
Введение	4
Глава 1. Интегральные операторы и машинное обучение	13
1.1 Машинное обучение в обработке изображений	13
1.2 Карты и вектора признаков	16
1.3 Интегральные преобразования и обработка изображений	20
1.3.1 Преобразование Радона	21
1.3.2 Преобразование Хафа	22
1.3.3 Быстрое преобразование Хафа	24
1.4 Интегральные преобразования и нейронные сети	26
1.5 Выводы. Постановка задачи	32
Глава 2. БПХ как слой нейронной сети	35
2.1 О рецептивных полях	35
2.2 Прямое БПХ	39
2.3 Диадические паттерны прямых в БПХ	44
2.4 Распространение градиента через БПХ	45
2.5 Транспонированное БПХ	47
2.6 Инвариантность к позиции в Хафе и Радоне	52
2.7 Сохранение информации	54
2.8 Результаты главы 2	57
Глава 3. Архитектуры сетей с БПХ и способы их применения	58
3.1 Полносвязная сеть со слоями БПХ	60
3.2 Архитектура Хафнет	62
3.3 Архитектура Хафэнкодер	65
3.4 Исследование возможностей Хафэнкодера	68

3.5	Влияние слоя преобразования Хафа на требуемое количество операций	71
3.6	Перенос знаний	71
3.7	Результаты главы 3	75
Глава 4. Практические применения нейронных сетей разработанных архитектур		77
4.1	Поиск внутренней точки схода полносвязной сетью с БПХ	77
4.2	Задачи, решаемые в процессе обработки изображений документов	80
4.2.1	Поиск внешних точек схода для документов	80
4.2.2	Семантическая сегментация изображений документов	85
4.2.3	Использование Хафэнкодера для поиска МЧЗ	88
4.3	Поиск внутренней точки схода на изображениях дороги	91
4.4	Практическое применение	95
4.4.1	Общее описание системы обучения	95
4.4.2	Система SmartID Reader	96
4.5	Результаты главы 4	98
Заключение		100
Список сокращений и условных обозначений		104
Список литературы		107
Список рисунков		117
Список таблиц		120
Приложение А. Акты о внедрении		121

Введение

Актуальность темы. Активное развитие области науки, связанной с нейронными сетями, привело к тому, что все большее количество задач решается с их использованием. При этом от создателей зачастую не требуется глубокого понимания внутренних принципов работы, что в совокупности с возросшими вычислительными мощностями персональных компьютеров и видеокарт сделало нейронные сети инструментом действительно массового пользования. В этих условиях критически важно проводить исследования, направленные на разработку методов создания оптимальных архитектур нейронных сетей и на понимание внутренней работы нейросетей. Первое важно, так как все большее количество алгоритмов, работающих на устройствах с все меньшими вычислительными мощностями и конечным запасом заряда батареи, используют нейронные сети. Второе необходимо для того, чтобы у исследователей и инженеров были механизмы построения архитектур, исходя из специфики задачи, что в свою очередь тоже позволит сократить вычислительные расходы. Рассматриваемые в работе задачи и методы их решения направлены на 1) создание оптимальных архитектур нейронных сетей для решения конкретных задач и 2) построения нейронных сетей с понимаемым принципом работы.

Искусственные нейронные сети появились в конце 1940-х годов. Тогда были разработаны первые модели, включая персептрон Розенблатта. И хотя сверточные слои использовались и раньше, например в работах ЛеКуна еще в 1998-м году, самый большой за всю историю скачок их развития начался сравнительно недавно, в 2006-м году. Именно тогда начали выходить работы Хинтона, Салахутдинова и Крижевского, посвященные построению многослойных конструкций из сверточных слоев. Эти слои принципиально отличались от полносвязных тем, что могли учитывать лишь информацию в локальной окрестности на картах признаков в противовес полносвязным слоям, которые всегда учитывали всю входную карту

признаков. Таким образом, у выходных нейронов сверточных слоев получается ограниченное рецептивное поле, что соответствует специфике задачи обработки изображений. Именно эта особенность подобных слоев и сыграла ключевую роль в их высокой эффективности в задачах обработки изображений.

Взросшие вычислительные мощности позволяют создавать большие нейронные сети, содержащие десятки и даже сотни миллионов обучаемых параметров, которые демонстрируют превосходное качество работы во многих задачах. Однако, в противовес этому, разработка нейросетевых архитектур и, особенно, новых типов слоев, которые позволили бы добиваться того же или сравнимого результата при значительно меньших вычислительных затратах – актуальная проблема. Например, введение слоев пулинга помогло увеличить обобщающую способность сверточных нейронных сетей путем понижения размера карты признаков, а последующее введение слоев, которые способны тем или иным способом увеличить изображение, породило целое семейство нейросетей типа “автокодировщик”, которые теперь повсеместно используются в задачах обработки, сжатия и улучшения изображений.

Однако, так или иначе, проблема рецептивного поля, т.е. влияния каждого входного нейрона на финальный результат, остается и решать ее “локальными” средствами совсем не просто. Разумеется, если мы знаем о систематических связях нейронов, находящихся далеко друг от друга на изображении, мы можем объединять их в “нелокальные” группы, что и используется в некоторых конкретных задачах.

С другой стороны, в задачах обработки изображений широко используется интегральное преобразование Хафа. Преобразование Хафа – дискретизованный вариант преобразования Радона, где каждое значение на преобразованной плоскости есть сумма значений вдоль соответствующей прямой исходной плоскости. Самое частое использование преобразования Хафа – выделение прямых: ясно, что если на изображении есть одна яркая прямая, то в соответствующей точке Хаф-образа значение будет максимальным.

Ввиду важнейшего значения, которое играют прямые на изображениях (дороги, дома, лучи томографа, границы документа, строки и т.п.), роль Хаф-анализа можно сравнить с ролью Фурье анализа в акустике. В то же время проведение Хаф-анализа изображения представляет собой довольно сложную задачу. На изображениях мы имеем дело не с прямыми, а с отрезками различной длины. Эти отрезки почти всегда не совсем прямые, часто зашумлены и/или частично не видны, поэтому Хаф-анализ изобилует алгоритмами эвристического характера. Этому посвящены, в частности, работы Ершова, Карпенко, П. П. Николаева.

Между тем, как раз с теми проблемами, которые затрудняют Хаф-анализ изображения, нейронные сети прекрасно справляются во многих задачах обработки изображений и, более того, существуют систематические методы их решения. Иногда нейронные сети используются непосредственно в Хаф-анализе, но в качестве исключительно обработчиков изображений.

Итак, в области обработки и анализа изображений известно множество алгоритмов, плодотворно использующих преобразование Хафа, но без нейронных сетей качество их работы в реальных условиях уже не удовлетворяет ожиданий. Поэтому “вплетение” преобразования Хафа в нейронные сети, причем не как поставщика первичных признаков, а в качестве полноценного слоя выглядит очень естественным.

Дополнительно стоит отметить, что использование преобразования Хафа переводит карты признаков в пространство проекций. Задача анализа пространства проекций встает при томографической реконструкции. Для реконструкции используются свертки и транспонированное преобразование проецирования, отображающее проекции в исходное пространство. Уже предлагаются нейросетевые модели, включающие соответствующие слои. Важно отметить, что здесь же возникает задача обращения подобного преобразования, которая может быть решена с использованием транспонированного преобразования. Например, в работе Адлера и Октема рассматривается глубокая end-to-end модель, которая содержит

слои трансформации карт признаков из координатного пространства в пространство признаков и обратно, а Сибен и соавторы для этого использовали целую комбинацию из слоев, решая аналогичную задачу. Таким образом, авторы этих и других работ решают схожую задачу: с использованием Хафовских слоев специального вида они пытаются создать архитектуру нейронной сети, способную решать конкретную задачу.

Однако для того, чтобы систематически сделать преобразование Хафа слоем нейронной сети, необходимо серьезное исследование, связанное с тем, как вести обучение сети при наличии таких слоев, а также с тем, в каких архитектурах использование будет эффективным. Таким образом, исследование, направленное на создание слоя преобразования Хафа и создание нейросетевых архитектур, содержащих такие слои, а так же исследование их свойств является актуальной задачей. Этой теме и посвящена диссертация.

Основные результаты были получены в процессе выполнения работа по следующим научным грантам РФФИ:

1. 15-29-06083 – “Исследования устойчивости методов статистического распознавания к искажениям в неконтролируемых условиях мониторинга аномалий земной поверхности в арктических широтах”
2. 17-29-03161 – “Каскадные алгоритмы локализации и отслеживания протяженных объектов на мобильных устройствах”
3. 17-29-07092 – “Методы машинного обучения для построения оптимальных тепловых карт в пространстве параметров элементов решения для комбинаторных задач”
4. 18-29-26027 – “Использование аппроксимаций операторов прямого и обратного проецирования арифметическими сетями низкой сложности в качестве слоев искусственной нейронной сети”
5. 19-29-09064 – “Исследование нейросетевых методов кооперации локальных и интегральных признаков для получения

карт информационной плотности при анализе изображений сцен”

6. 19-29-09092 – “Модели и методы компьютерного зрения в задачах обеспечения информационной безопасности для индустрии 4.0”

Целью данной работы является исследование свойств нейронных сетей, включающих слои прямого и транспонированного преобразований Хафа при решении прикладных задач компьютерного зрения.

Для достижения этой цели поставлены и решены следующие **задачи**:

1. Исследование трансформации рецептивного поля при добавлении слоя быстрого преобразования Хафа. Это необходимо для мотивированного построения нейросетевых архитектур при решении прикладных задач.
2. Изучение способов и особенностей распространения градиента через слои быстрого преобразования Хафа. Без такого способа обучение нейронной сети в end-to-end режиме невозможно. Оценить вычислительную сложность и практическую применимость.
3. Создание семейства нейросетевых архитектур для решения набора практических задач. Данные архитектуры должны быть объяснимы с точки зрения классического использования интегральных операторов.
4. Исследование поведения сетей предложенных архитектур при решении ими прикладных задач компьютерного зрения: поиск внутренней точки схода, поиск внешней точки схода, детекция выпуклых объектов. Показать, что нейросети разработанных архитектур обеспечивают конкурентное качество при меньшем количестве обучаемых параметров и/или при требовании меньшего количества вычислительных мощностей.

Научная новизна:

1. В работе впервые проведено исследование трансформации рецептивного поля нейронной сети при добавлении в нее слоев прямого и транспонированного быстрого преобразования Хафа. Показано, что для того, чтобы нейронная сеть была способна решать задачи компьютерного зрения, ее рецептивное поле должно обладать достаточным размером.
2. В работе предложена новая архитектура сетей со слоями прямого преобразования Хафа, а также метод ее использования для решения задачи поиска внешней точки схода.
3. В работе впервые предложена архитектура нейронной сети с использованием транспонированного преобразования Хафа для решения задач компьютерного зрения и показано, что такая архитектура способна решать задачи поиска внутренних точек схода и семантической сегментации четырехугольных объектов с превосходным качеством, при этом обладая на два порядка меньшим количеством обучаемых параметров.
4. При применении известной технологии переноса знаний для domain adaptation выявлен новый эффект: перенос знаний успешно происходит, но для высокоуровневых, а не для низкоуровневых признаков.

Практическая значимость работы подтверждается тем, что методы обучения нейронных сетей с использованием исследованных в работе слоев преобразования Хафа были внедрены в систему обучения нейронных сетей, используемую в компании ООО “Смарт Энджинс Сервис”. Это позволило внедрить предложенные архитектуры в уже имеющиеся решения для обработки изображений и распознавания документов, которые интегрированы в информационную инфраструктуру ряда коммерческих организаций, а также в ряд информационных решений государственных структур Российской Федерации.

Соответствие диссертации паспорту научной специальности. В соответствии с формулой специальности 1.2.2 “Математическое моделирование, численные методы и комплексы

программ” (технические науки) в работе выполнено исследование слоя преобразования Хафа, приведено доказательство того, что транспонированный оператор для быстрого преобразования Хафа выполняется с той же сложностью, что и прямой, разработаны архитектуры с такими слоями и исследованы их свойства. Работа соответствует следующим пунктам паспорта специальности: п. 2 “Разработка, обоснование и тестирование эффективных вычислительных методов с применением современных компьютерных технологий”, п. 3 “Реализация эффективных численных методов и алгоритмов в виде комплексов проблемно-ориентированных программ для проведения вычислительного эксперимента”, п. 7 “Качественные или аналитические методы исследования математических моделей (технические науки)”.

Методология и методы исследования. В работе используются методы цифровой обработки изображений, методы стохастической непрерывной оптимизации, методы статистического анализа. Все результаты обосновываются формально-дедуктивно либо вычислительным экспериментом.

Основные положения, выносимые на защиту:

1. Введение слоев прямого и транспонированного быстрого преобразования Хафа позволяет сделать рецептивным полем выходных нейронов все входное изображение без увеличения числа обучаемых параметров.
2. При помощи представленной архитектуры Хафнет можно построить конечную карту признаков, которая с помощью кусочно-проективных преобразований отображается в бесконечную плоскость, содержащую исходное изображение, что позволяет работать в том числе и с бесконечно удаленными точками.
3. Использование комбинации прямого и транспонированного преобразований Хафа в нейронной сети позволяет построить автокодировщик, способный учитывать интегральные признаки, но не имеющих дополнительных обучаемых параметров.

4. При обучении блоков сверточных слоев, разделенных слоями преобразования Хафа для решения задачи поиска точки схода, накопление специфичных для конкретных изображений признаков происходит в первом блоке сверточных слоев, в то время как второй и третий блоки можно использовать для решения той же задачи на данных из другого домена.

Достоверность полученных результатов подтверждается соответствием теоретических и экспериментальных результатов, продемонстрированных в работе, успешной апробацией результатов и внедрением в коммерческие системы распознавания документов.

Апробация работы. Основные результаты работы докладывались и обсуждались на следующих конференциях:

1. Международной конференции “International Conference on Machine Vision” (ICMV) в 2016-м, 2017-м и 2019-м годах.
2. Международной конференции “International Conference on Document Analysis and Recognition” (ICDAR) в 2019-м году.
3. Международной конференции “International Conference on Image Processing” (ICIP) в 2020-м году.

Личный вклад. Все основные результаты, изложенные в данной работе, были получены лично автором. Постановка задач и обсуждение результатов проводились совместно с научным руководителем. В [7] автором представлено доказательство выполнимости транспонированного преобразования Хафа с такой же асимптотической сложностью, как и быстрое прямое преобразование Хафа. В [3, 4] анализируется преобразование Хафа в качестве слоя нейронной сети для решения задач поиска внутренней точки схода и классификации изображений, содержащих линии. В [5] автором предлагается архитектура Хафнет и способ ее применения для решения задачи поиска внешних точек схода на изображениях, содержащих документы. В [2, 6, 7] автором предлагается архитектура Хафэнкодер, где решаются задачи поиска машиночитаемой зоны, поиска внутренней точки схода и семантической сегментации. В [1] автор исследует проблему переноса знаний между данными из

различных доментов для сети архитектуры Хафнет. В этой работе рассматривается два домена (изображения документов и изображения дорог с видео регистраторов) и показывается, что накопление знаний происходит в слоях, идущих после преобразования признаков в пространство (s, t) . В [8] выполнено исследование слоя быстрого преобразования Хафа в задаче классификации изображений и опубликовано без соавторства. В [9,10] приводятся ссылки на свидетельство о регистрации программы для ЭВМ и патент США.

Публикации. Основные результаты по теме диссертации опубликованы в 8 печатных изданиях, 1 из работ изданы в журналах, рекомендованных ВАК, 7 – в изданиях, индексируемых в Web of Science и Scopus. Также получено свидетельство о регистрации программы для электронных вычислительных машин (ЭВМ) и патент США.

Объем и структура работы. Диссертация состоит из введения, четырех глав, заключения и одного приложения. Полный объем диссертации составляет 123 страницы, включая 38 рисунков и 9 таблиц. Список литературы содержит 95 наименований.

Глава 1. Интегральные операторы и машинное обучение

1.1 Машинное обучение в обработке изображений

Машинное обучение - раздел науки, изучающий алгоритмы, которые могут улучшать свои характеристики по мере накопления данных [1]. Такие алгоритмы широко используются в тех случаях, когда четкая математическая постановка задачи невозможна или слишком трудна, но существует способ получить обучающие данные в достаточном объеме. Необходимый объем данных сильно зависит и от задачи, и от используемого алгоритма. Такие алгоритмы естественным образом подходят для решения задач предсказания самых разных событий по накопленной статистике, будь то банкротство банка, погода или заболеваемость людей [2, 3, 4]. Кроме этого, задачи обработки изображений, такие как сегментация, распознавание, тэгирование и прочие, легко решаемые человеком естественным образом, подходят для создания обучающего набора данных, хотя точного алгоритма их решения мы не знаем и относим эти задачи к сфере искусственного интеллекта.

В реальности к методам машинного обучения относят не только случаи, когда алгоритм изменяется, накапливая опыт в процессе функционирования, но и случаи, когда алгоритм настраивается “одноразово”, имея перед собой набор примеров, где известны входные данные и нужный ответ, а задача состоит в поиске параметров, при которых выход алгоритма был бы максимально приближен к ответу.

В этом смысле машинное обучение практически не отличается от давно известной задачи идентификации параметров, где речь идет об устройствах, схема работы которых зависит от неизвестных параметров. Наблюдая выходы устройства при различных входах, мы можем попытаться определить значения параметров.

В обоих случаях постановка задачи выглядит следующим образом. Имеется функция (алгоритм, устройство), которая преобразует входные данные X в выходные Y : $Y = F(X, W)$, где $X = (x_j), j \in [0, J)$ – вектор входных данных, $Y = (y_k), k \in [0, K)$ – вектор выходных данных, $W = (w_0, \dots, w_{M-1})$ – вектор неизвестных параметров. Также дан набор примеров $\{\langle X_i, Y_i \rangle\}, i = 0, \dots, N-1$. Требуется найти такие значения $\bar{W}$, чтобы для $\bar{Y}_i = F(X_i, \bar{W})$ значение рассогласования было минимальным. Разумеется, рассогласование должно быть определено. К примеру, рассогласование S можно определить следующим образом 1.1:

$$S(\bar{W}) = \sum_i |Y_i - F(X_i, \bar{W})|. \quad (1.1)$$

Так или иначе, мы имеем задачу поиска значений $\bar{W}$, дающих минимум функции S на данном множестве.

Итак, задачи машинного обучения и идентификации параметров полностью совпадают. Разумеется, в задаче идентификации параметров никто не запрещает корректировать параметры устройства при получении новых данных, как и в машинном обучении. Вопрос только в наличии эффективных алгоритмов и реального времени на такую корректировку. В то же время практика решения этих двух задач существенно различна. В задачах идентификации параметров функция $F(X, W)$ обычно достаточно сложна (например, описывается дифференциальными уравнениями), число параметров M невелико, а параметры уникальны. Напротив, количество параметров в задачах машинного обучения часто составляет миллионы и миллиарды. При этом сами параметры разбиваются на сравнительно небольшое число однородных групп.

При всех различиях практик не следует забывать о полной идентичности математических постановок и о возможности переносить некоторые методы, а иногда и результаты из одной области в другую. В частности, в некоторых задачах речь может идти не только об идентификации параметров, но и об “идентификации

модели” (например, регулятора), когда функция $F(X, W)$ может менять свою структуру.

Лидирующие позиции в решении многих задач искусственного интеллекта на данный момент занимают алгоритмы, которые так или иначе включают в себя нейронные сети, а иногда и исключительно из них состоящие. Здесь рассматриваются самые разнородные задачи: предсказание событий, семантическая сегментация [5], бинаризация [6], классификация изображений и многие другие. В последнее время с помощью нейронных сетей стали решать еще и задачи генерации данных [7] и даже атак на другие нейронные сети [8, 9].

Причин тому несколько, и основную выделить тяжело. Сильно возросшие вычислительные мощности процессоров и видеокарт, относительная универсальность и удобные общедоступные инструменты для обучения в совокупности с наличием заметного количества обучающих наборов данных в открытом доступе развязывают руки исследователям. А “модульность” нейронных сетей позволяет исследователям и разработчикам использовать нейронные сети в своих алгоритмах, не вдаваясь в подробности их устройства. Современные инструменты, имеющиеся в открытом доступе, не требуют от разработчика или исследователя хоть сколько-то глубокого понимания принципов работы и обучения нейронных сетей, но позволяют их более или менее успешно использовать в качестве блоков в алгоритмах более высокого уровня. Это привело к некоторому “замусориванию” базы публикаций на тему решения задач при помощи нейронных сетей, так как появилось очень большое количество публикаций, в которых предлагаются те или иные модификации без объяснения причин, почему это должно было сработать и какие именно механизмы привели к улучшению целевых показателей.

Вообще, смешение гипотез, точно поставленных и воспроизводимых экспериментов и чисто инженерных “открытий” (“мы сделали так и получилось хорошо”), чаще всего не сопровождающихся точным описанием, очень характерно для нынешнего

огромного массива публикаций, посвященных нейросетевым алгоритмам. Для таких инженерных публикаций типично наличие в моделях скрытых/неописанных параметров, изменение которых может заметно влиять на их итоговое качество. Существуют даже работы, опубликованные в которых результаты прямо противоречат друг другу. В работе [10] приводятся примеры таких противоречащих друг другу публикаций на примере задачи построения описателя изображения.

Тема нейронных сетей очень хорошо покрыта литературой, как со стороны теории, так и со стороны практики [11]. Тем не менее, хотелось бы обратить внимание на один важный момент: популярность нейронных сетей носит скачкообразный характер. Согласно общепринятому мнению имело место три выраженных волны, две из которых показаны на Рис. 1.1, взятому из книги [12]. Первая волна приходится на 1940 — 1960-е годы. Именно тогда были разработаны первые модели, в том числе перцептрон Розенблатта [13]. Вторая волна началась в 1980-е годы и была связана с обретением нейронными сетями глубины. Тогда же метод обратного распространения ошибки был применен для обучения модели с одним или двумя скрытыми слоями [14]. Третья волна началась сравнительно недавно, в 2006-м году [15, 16], и продолжается до сих пор. Это волна “глубокого обучения” и сверточных нейронных сетей. В последнее время существует тенденция называть “глубоким обучением” только те алгоритмы, которые созданы для работы на исходных данных, а не на инженерно-вычисленных признаках.

1.2 Карты и вектора признаков

Прежде чем перейти к рассмотрению структуры нейронных сетей и особенностей поведения тех или иных архитектур, стоит сосредоточиться на следующем вопросе: какими сущностями оперируют нейронные сети в целом и отдельные слои в частно-

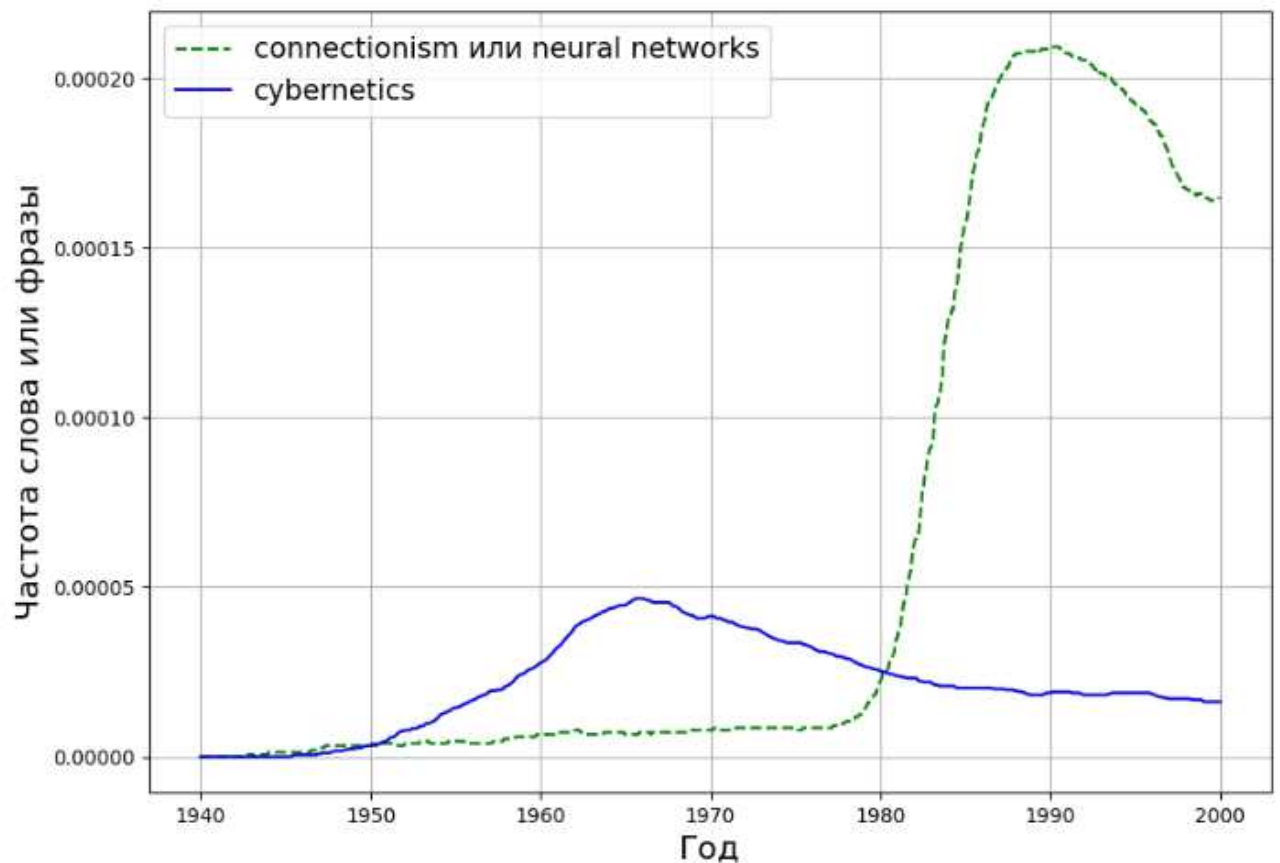


Рисунок 1.1 — Две из трех волн развития нейронных сетей.

сти? Так как все детали современных нейронных сетей подробно описаны в огромном количестве источников, одним из самых популярных среди которых является книга Гудфелоу [12], рассмотрим лишь некоторые понятия, необходимые для дальнейшего изложения. Если во входных данных можно задать систему координат как, например, на изображении, то такие данные будем называть картой признаков (feature map). Размерность такой карты варьируется в зависимости от типа данных, с которыми идет работа. Для последовательностей сигналов эта размерность может быть единицей, для изображений двойкой, для пространственных изображений тройкой. Кроме этого, подобные карты могут быть многоканальными, т.е. может иметься много карт с одинаковой системой координат. Входом и выходом сверточных слоев нейронной сети, чаще всего, являются карты признаков. Если все элементы входных данных сети или слоя нельзя уложить в систему координат, то тогда это называется вектор признаков (feature vector). С

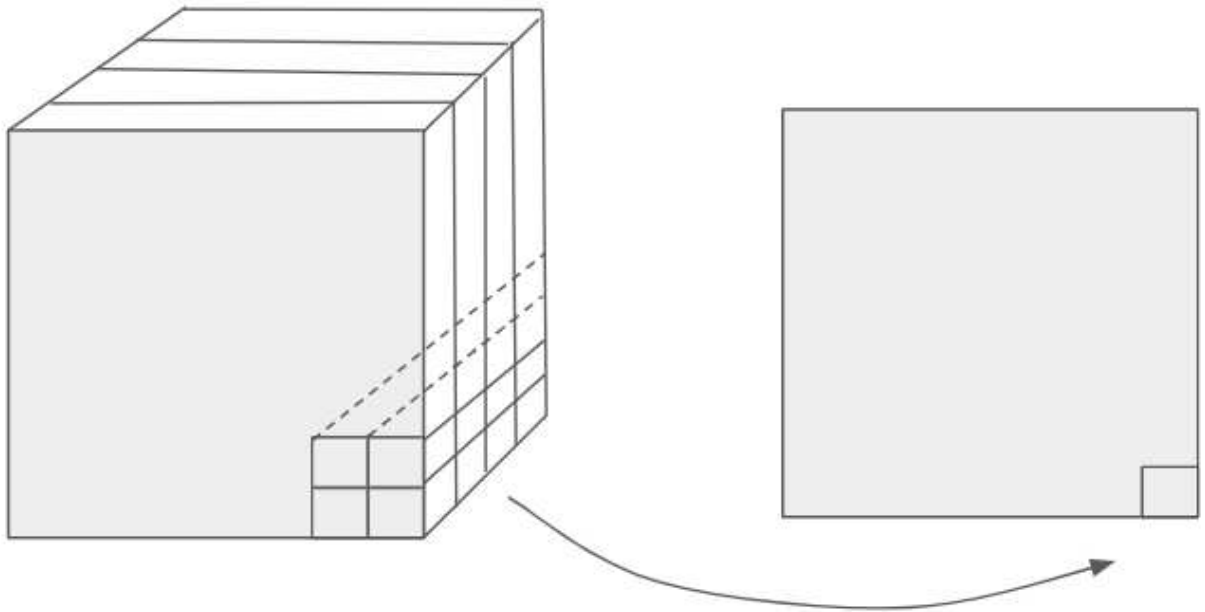


Рисунок 1.2 — Применение ядра сверточного фильтра размером 2×2 к многоканальной карте признаков.

подобной структурой данных, как правило, работают полносвязные слои. Легко заметить, что любая карта признаков является вектором признаков, но не наоборот. Элементы карты признаков в двумерном случае называются пикселями, а в трехмерном – вокселями. Параметры двумерных карт признаков обычно задаются тремя значениями: ширина, высота и количество. Таким образом, обозначение $10 \times 10 \times 32$ говорит о том, что карта имеет 10 пикселей в ширину и в высоту, а количество карт равно 32.

Нейронные сети состоят из различных слоев и функций активаций, которые тоже представляются в виде отдельного слоя, но все-таки имеют принципиальное отличие. Самые популярные виды обучаемых слоев - сверточный слой и полносвязный слой. Каждый выход этих слоев является взвешенной суммой некоторой области входного вектора признаков, как показано на Рис. 1.2, и, следовательно, сам слой является линейным преобразованием входного вектора. Для того, чтобы последовательные линейные слои нельзя было свести к единственному линейному преобразованию, между ними всегда ставят нелинейную функцию активации, которая делает такое схлопывание невозможным. Принципиальным отличием

сверточного слоя от полносвязного является часть входа, по которой считается результат: для сверточного слоя это, как правило, небольшое окно на входной карте признаков, а для полносвязного – весь вектор признаков. Формально, любой сверточный слой можно в точности представить в виде полносвязного, но, тем не менее, именно появление жесткого ограничения на локализацию признаков дало очередной толчок к развитию глубокого обучения [17]. Это произошло сразу по нескольким причинам. Во-первых, в такое ограничение неявно встроено требование к выделению локальных признаков, во-вторых, количество обучаемых параметров сильно сокращается. Таким образом в архитектуру нейронной сети зашивается “сложная регуляризация”, которая не просто требует от сети специфического поведения (выделения локальных признаков в данном случае), а забирает у нее возможность работать иначе. Еще одним немаловажным фактом является то, что при обучении параметров сверточного фильтра каждая точка приложения порождает градиент. В результате получается, что для данного фильтра количество эффективных обучающих примеров равно количеству приложений фильтра, что существенно больше, чем количество изображений. В самом деле, при обучении сверточного фильтра сначала вычисляется градиент обучаемых параметров для каждой точки приложения фильтра по отдельности, затем все это суммируется и лишь потом происходит обновление значений параметров.

Отдельно выделим класс нейронных сетей, которые состоят только из слоев, вычисление выхода которых происходит исключительно по некоторому конечному окну входной карты признаков. Такие сети называются полносверточными, и их отличительной особенностью является то, что они могут быть применены к изображениям любого размера (с минимальным размером, задаваемым конфигурацией слоев). Выходом такой сети являются карты признаков, а значения в них несут смысловую нагрузку, заданную функцией потерь и разметкой.

1.3 Интегральные преобразования и обработка изображений

В задачах обработки изображений на данный момент существует множество инженерных алгоритмов, которые включают в себя различные интегральные преобразования. Сюда входят и алгоритмы обработки изображений, выходом которых тоже является изображение, но обладающее дополнительными свойствами (отсутствием шума, бинарной структурой и т.п.), и алгоритмы поиска на изображении различных объектов. Главными преимуществами таких алгоритмов обычно являются скорость работы и понятность принципа действия и, следовательно, предсказуемость поведения [18, 19, 20, 21].

Например, в работе [22] описано, как можно использовать дискретизованное преобразование Радона для поиска точек схода на документе. Авторы предлагают алгоритм, в котором сперва выделяют границы объектов на изображении (без нейронных сетей) и делают преобразование Радона, а затем в пространстве Радона ищут локальные максимумы, которые соответствуют прямым линиям на исходном изображении. Линии, которые пересекаются в одной точке, порождают максимумы, лежащие на одной прямой. Здесь подразумевается, что документ содержит достаточно элементов с прямолинейными параллельными границами, продолжения которых порождают как минимум две точки схода: горизонтальную и вертикальную.

Заметим, что выделение границ, в том числе и прямолинейных, на изображении – задача, которая хорошо решается полносверточными нейронными сетями [23], а поиск точки схода с использованием преобразования Радона – выбор максимумов после предобработки изображения (подавление шумов, ложных максимумов и т.п.), т.е., после операции, которую нейросеть способна делать хорошо [24, 25].

Этот и множество других примеров подталкивают к очевидному вопросу: а можно ли создать алгоритм, который бы обладал

преимуществами обоих методов? Объединение подходов напрашивается само собой, потому что проблемой классических алгоритмов является плохая устойчивость к неидеальности входных данных, в это же время, нейросетевые автоэнкодеры хорошо себя зарекомендовали в задаче обработки изображений.

В данной работе в качестве интегрального преобразования будет рассмотрено преобразование Хафа. Это преобразование является дискретизацией преобразования Радона, в которой параметризации прямых могут несколько различаться в зависимости от реализации. Рассмотрим эти преобразования и их смысл подробнее.

1.3.1 Преобразование Радона

Преобразование Радона – интегральное преобразование функции многих переменных, которое впервые было введено Иоганном Радонам в 1917-м году. Описываемое в непрерывном пространстве для функции двух переменных $f(x, y)$, это преобразование обладает простым геометрическим смыслом – это интеграл от функции вдоль прямой, перпендикулярной вектору $n = (\cos \varphi, \sin \varphi)$ и проходящей на расстоянии ρ от начала координат.

$$R(\rho, \varphi) = \int_{-\infty}^{\infty} f(\rho \cos \varphi - z \sin \varphi, \rho \sin \varphi + z \cos \varphi) dz \quad (1.2)$$

На Рис. 1.3 проиллюстрировано применение формулы 1.2. Вообще говоря, преобразование Радона имеет большое значение в физике и, в частности, в компьютерной томографии, так как получение набора проекций объекта при постепенно меняющемся угле проецирования очень сходно с вычислением преобразования Радона от объекта, а решение задачи компьютерной томографии связано с обращением преобразования Радона.

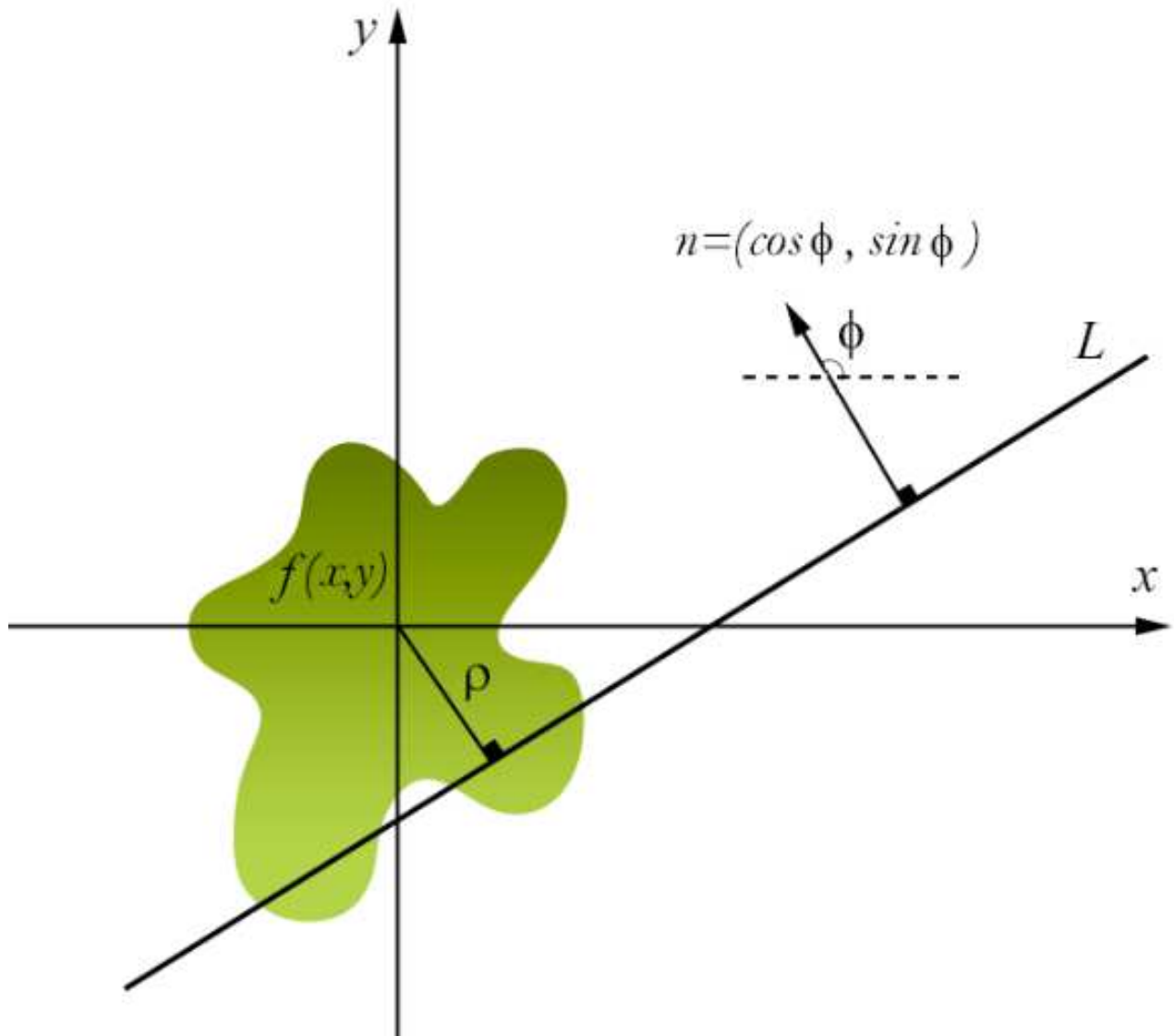


Рисунок 1.3 — Вычисление преобразования Радона от функции двух переменных.

1.3.2 Преобразование Хафа

Преобразование Хафа в двумерном случае можно рассматривать как дискретизацию преобразования Радона. В работе [26] подробно описаны оба преобразования.

Стоит оговориться, что само по себе преобразование Хафа подразумевает суммирование не только по прямым, но и по различным геометрическим примитивам (эллипсы, окружности...), а обобщенное преобразование Хафа, предложенное Баллардом в 1981 году [27], допускает возможность использования шаблонов и бо-

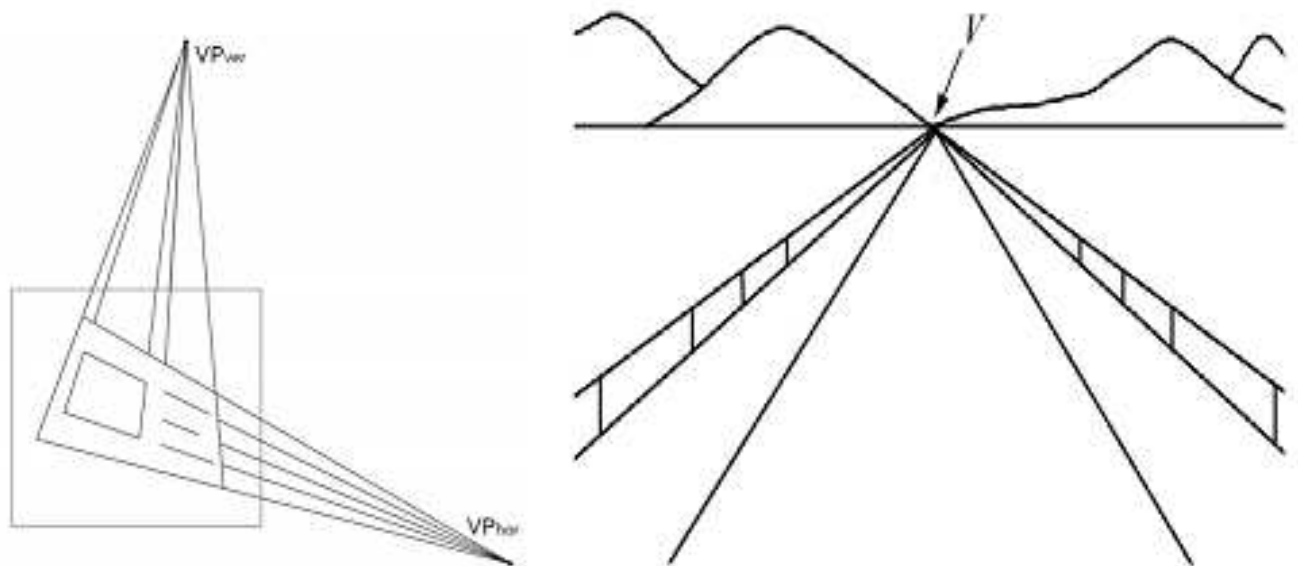


Рисунок 1.4 — Несобственные (слева) и собственная (справа) точки схода.

лее общей формы. Таким образом, результатом преобразования Хафа является множество значений выбранной операции для пикселей (обычно это сумма интенсивностей), подходящих под шаблон. Размерность такого результата определяется количеством параметров шаблона и для случая суммирования по прямым равно двум. Значение в каждой точке результирующего пространства может рассматриваться, как количество голосов за то, что именно заданный примитив с такими параметрами присутствует на исходном изображении. Отсюда получаются голосовательные схемы, ответом которых являются положение и параметры искомого объекта.

В данной работе мы будем рассматривать преобразование Хафа не как способ поиска объектов, а как операцию над изображением, результатом которой тоже является изображением, но в других координатах. В случае интегрирования по прямым, который и будет рассматриваться, преобразование Хафа является набором сумм интенсивностей пикселей на исходном изображении, уложенных в координатную систему, где по осям отложены параметры прямых. Вычисление такого преобразования тривиальным алгоритмом имеет сложность $O(n^3)$, где n - линейный размер входного изображения.

Как уже было сказано, преобразования Хафа часто используются в алгоритмах поиска прямых или отрезков на изображении.

Существуют и алгоритмы, содержащие в себе последовательные применения такого преобразования. В обработке изображений классическим примером алгоритма, использующего двойное преобразование Хафа, является поиск точки схода на изображении. Для того, чтоб объяснить, что такое точка схода необходимо представить две параллельным прямые в трехмерном пространстве и спроецировать его на проективную плоскость. Точка пересечения таких прямых на этой плоскости и будет точкой схода. Такая точка может находиться как в центре изображения, так и быть бесконечно удалена. Разделяется два разных случая: точка схода лежит внутри изображения (собственная) или снаружи (несобственная). На рисунке 1.4 показаны несобственные, образованные продолжениями сторон исходно прямоугольного объекта, и собственная, образованная пересечением продолжений краев дороги.

Отдельным направлением является не просто использование преобразования Хафа, но его обучение. В работах [28, 29] авторы представляют нейросетевые модели для более тонкой настройки шаблонов, по которым делается операция. Кроме этого, такой подход позволяет более или менее удобно формализовать конкретное применение обобщенного преобразования Хафа и объединить его с обученными порогами для построения голосовательной схемы.

1.3.3 Быстрое преобразование Хафа

Для вычисления сумм по различным прямым на изображении существует быстрый алгоритм или быстрое преобразование Хафа (БПХ в дальнейшем), вычисление которого имеет асимптотическую сложность $O(n^2 * \log(n))$. Этот алгоритм был впервые предложен в 1992-м году Брэйди [30]. Ускорение достигается за счет повторного использования частичных сумм по отрезкам прямых, но для этого необходимо, чтобы прямые состояли из рекуррентных паттернов. Такое представление приводит к тому, что с увеличени-

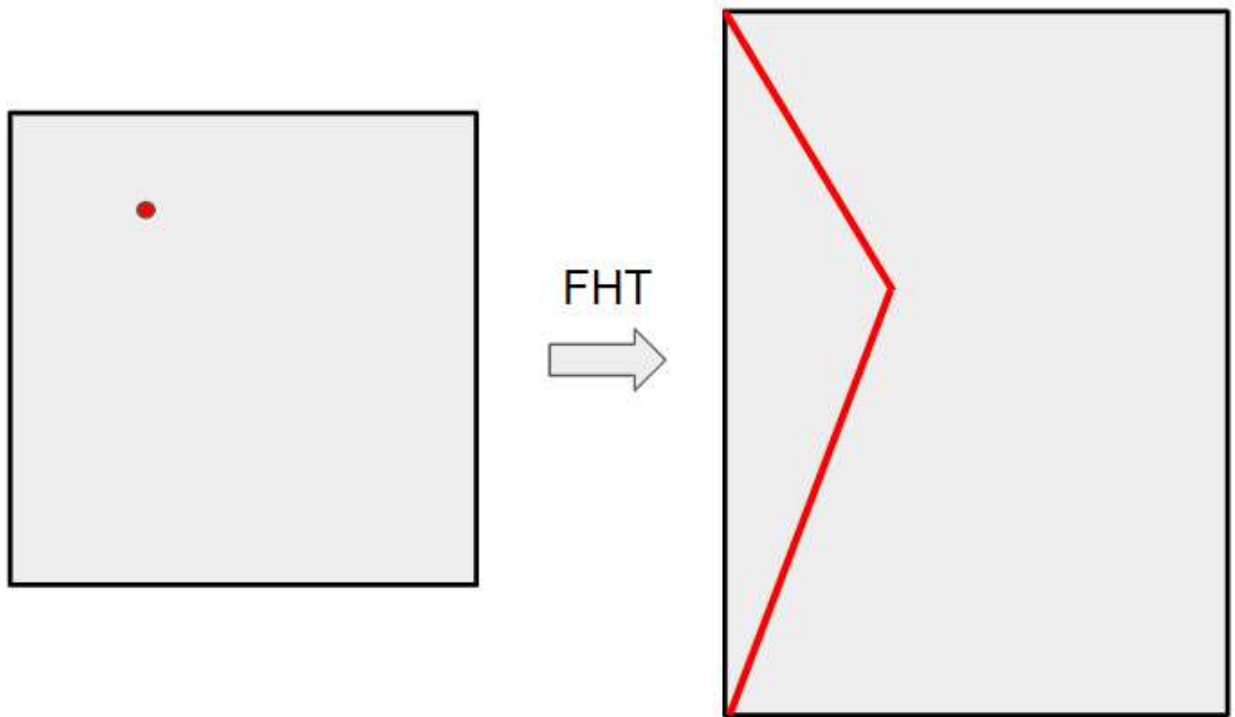


Рисунок 1.5 — Быстрое преобразование Хафа от изображения с единственной точкой. Схематичное изображение.

ем размера изображения накапливается ошибка [31]. Доказано, что эта ошибка с ростом размера входного изображения растет лишь логарифмически и во многих алгоритмах оказывается несущественной.

Использование данного алгоритма имеет некоторые особенности, одной из которых является наличие “излома” при переходе от преимущественно горизонтальных к преимущественно вертикальным прямым. В работе [32] доказано, что результат выполнения БПХ по всему диапазону углов будет иметь как минимум один такой “излом”. На рисунке 1.5 приведен результат преобразования по всем углам от изображения с единственной точкой. Можно видеть, что результатом является не прямая, а ломаная.

Подробнее быстрое преобразование Хафа будет рассмотрено в главе 2, где будет дано четкое определение рекуррентных паттернов, а так же выписаны формулы перехода точек в прямые, прямых в точки и доказаны некоторые свойства этого преобразования.

1.4 Интегральные преобразования и нейронные сети

Современные нейронные сети относительно успешно решают поставленные задачи, но иногда обладают чрезмерной сложностью. В некоторых задачах используются различные регуляризации и искусственные ограничения для того, чтоб отсечь заведомо неверные решения и упростить модель.

В случае с применением интегральных операторов задать явное ограничение на поведение сети можно путем использования линейного оператора какого-нибудь интегрального преобразования между слоями, тем самым вынудив нейронную сеть оперировать строго определенным видом признаков. Тем не менее, использование предопределенных линейных преобразований в нейронных сетях в явном виде – довольно экзотическая тема, которая редко исследуется. Возможно, это частично связано с тем, что в глубоких нейронных сетях, если нужно, такую роль способны выполнять полносвязные слои. Однако, если рассматривать обработку больших изображений (размером больше 100×100 пикселей) при помощи нейронных сетей, такие подходы теряют практическую применимость. Во-первых, полносвязные слои становятся недопустимо большими. Например, для карты признаков $100 \times 100 \times 64$ такой слой бы имел примерно $4 * 10^{11}$ обучаемых параметров, что сравнимо с крупнейшими из известных сетей [33, 34] целиком и неприемлемо для одного слоя. Во-вторых, как следствие, растет риск переобучения. В-третьих, глубокие полносверточные архитектуры благодаря количеству слоев могут набирать значительный размер рецептивного поля. Это приводит к тому, что подобные сети способны оперировать информацией с широкой зоны входного изображения и принимать во внимание признаки крупного размера, тем самым, снижая качественное различие с сетью, в которой интегральный оператор присутствует в прямом виде. Однако такая сеть будет громоздкой и ее можно будет сделать легче, меньше и

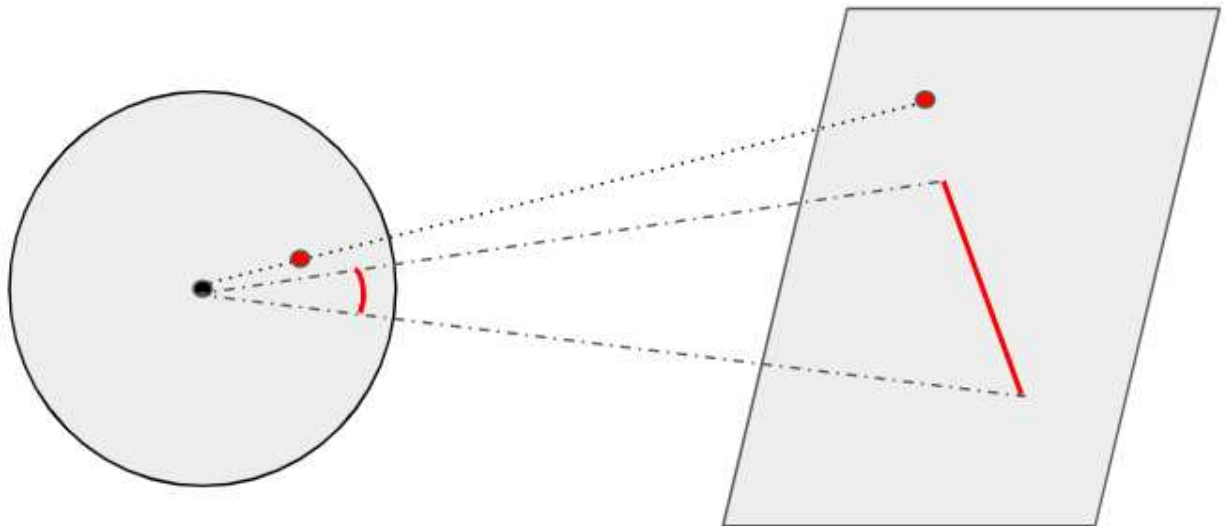


Рисунок 1.6 — Схема обратного гномонического проецирования.

понятнее, если использовать корректный интегральный оператор, если это, конечно, подходит для решаемой задачи.

Существуют работы, которые не используют преобразование Хафа в явном виде, но, тем не менее, используют проецирование на сферу в обобщенных координатах. Мы рассмотрим далее целый класс работ, представляющих различные модификации алгоритмов, которые в общих чертах можно описать тремя основными шагами: найти границы на исходном изображении, спроецировать на полусферу, найти точки схода на проекции. Практическая применимость таких методов существенным образом зависит от их алгоритмического состава, но, в целом, они страдают от недостаточной устойчивости к различным неидеальностям входных данных: яркостный или геометрический шум, наличие противоречивой информации на изображении, которые всегда в той или иной мере присутствуют в изображениях реальных сцен.

Например, в работе [35] авторы сначала инженерным методом находили прямые на изображении (использовался алгоритм, включающий в себя интегральный оператор над изображением). После этого проецировали изображение методом, похожим на обратное гномоническое проецирование как показано на Рис. 1.6, и далее, уже на гауссовой сфере, учили нейронную сеть. В работе показано, что преимуществом такого подхода является возможность

работать на разнородных данных, так как на гауссовой сфере уже нет признаков, специфичных задаче. Кроме этого, шаг проецирования позволяет перевести бесконечное исходное пространство в конечную область и облегчить работу с бесконечно удаленными точками. Однако они же признают, что такой подход имеет слабость: метод выделения исходных прямых сильно зависит от данных, требует тонкой настройки. В работе [36], наоборот, авторы применяют нейронную сеть для детекции границ на изображении и лишь затем используют преобразование Хафа и инженерные методы для получения финального результата.

В работе [37] описано использование оператора проецирования на сферу в обобщенных координатах прямых, предварительно найденных с помощью полносверточной нейронной сети. После этого уже к полусфере применяется “коническая” свертка, фильтры которой специальным образом применяются в “коническом пространстве”, где прямая в исходном изображении может быть определена локальными признаками. Тем самым, получается некоторое интегральное преобразованием исходного изображения. Оставшаяся часть схемы представляет из себя еще одну нейронную сеть. И хотя авторы утверждают, что вся модель может быть обучена end-to-end, в работе не описано способов передачи градиентов через отдельные ее элементы.

В работе [38] представлен похожий алгоритм, целью которого является не сам поиск точки схода, а поиск матрицы проективного преобразования для ректификации изображения. Авторы описывают способ получения этой матрицы, исходя из известной линии горизонта и вертикальной точки схода, которые, в свою очередь, ищут с помощью нейросетевой предобработки изображения и использования проецирования.

Нельзя не упомянуть работы, в которых описываются методы нахождения точки схода и без использования интегральных операторов. Идейно понятно, что хоть и существуют изображения, на которых достаточно лишь небольшой окрестности точки схода для ее нахождения, такие методы не могут работать в общем случае.

Например, в [39] утверждается, что поиск собственной точки схода возможно осуществить обычной сверточной нейронной сетью, основанной на архитектуре AlexNet. Такой подход сталкивается сразу с рядом проблем: последние полносвязные слои имеют огромное (в сравнении с остальной частью сети) количество обучающихся параметров и склонны к переобучению, точка схода не обязательно может быть найдена исходя из локальных признаков. Более того, в работе [40] с помощью нейронной сети и без использования каких-либо интегральных операторов находятся бесконечно удаленные точки схода. На выход нейронной сети применяется функция активации $\tanh$ и принимается, что чем ближе значение по модулю к единице, тем ближе оно к бесконечности. Такой подход сталкивается со всеми теми же трудностями и еще и с дополнительными. Необходимо исследовать устойчивость ответа, особенно для существенно удаленных точек схода.

Другим классом алгоритмов, включающих интегральные преобразования и нейронные сети, являются алгоритмы, в которых само преобразование применяется к исходному изображению и на его результате уже учится нейронная сеть. В работе [41] таким образом строится часть системы идентификации водителя, а в работе [42] – фигуры языка жестов.

В работе [43] рассматривается алгоритм для поиска дефективных объектов, состоящий из еще большего количества шагов. Так, авторы сперва предлагают сделать инженерный препроцессинг для уменьшения уровня шума. После этого, с помощью преобразования Хафа найти нужную зону на изображении для уменьшения фактической площади изображения и, следовательно, ускорения дальнейшей нейронной сети.

Более далекий от рассматриваемой темы класс алгоритмов, использующих обобщенное преобразование Хафа, включает в себя различные методы узнавания паттернов. Например, в работе [44] приведен алгоритм распознавания языка жестов. Согласно описанию, исходное изображение и карта глубины подаются на вход обобщенному преобразованию Хафа. Затем, выход Хафа и карта

глубин используются как входы на специально обученные сверточные нейронные сети, которые, в конечном итоге, классифицируют показанный жест, используя заранее определенный “алфавит”.

И хотя преобразование Фурье в данной работе не будет рассматриваться, но оно тоже является интегральным и невозможно не отметить его применимость в соответствующих задачах. Исследованию этих алгоритмов можно посвятить множество отдельных работ, а здесь лишь для общности картины упомянем некоторые из них.

В работе 1999-го года [45] авторы описывают способ учить полносвязную нейронную сеть на выходе преобразования Фурье от показателей ЭКГ для определения аритмии в реальном времени. Представленный алгоритм сам по себе прост и, согласно результатам, хорош в работе. В 2011-м году был опубликован схожий алгоритм [46], но существенно усложненный. Тут уже добавилось шумоподавление, преобразование Фурье заменилось на быстрое преобразование Фурье, а ответ сети стал не бинарным, а многоклассовым. В данных работах производилось одномерное разложение Фурье, в то время как в работе [47] используется двумерное разложение сигналов, полученных от вибраций двигателей. После разложения и специальной предобработки применяется вполне стандартная сверточная нейронная сеть для определения наличия аномалий в поведении двигателя.

Кроме этого, исследователи не ограничиваются исходными данными с заведомо присутствующей периодической составляющей. Так, в работе [48] авторы приводят способ обучать классифицирующую сеть в Фурье пространстве с целью ускорения. Утверждается, что такой трюк позволяет побороть экспоненциально растущую с размером входного изображения длительность обучения без потери качества. Однако в работе приведены только эксперименты на датасетах CIFAR10 (32×32) и MNIST (28×28), которые содержат в себе картинки небольших размеров.

Еще одним интересным применением преобразования Фурье в нейронных сетях является решение задачи обратной проекции,

возникающей в компьютерной томографии. В работе [49] описано разложение алгоритма FBP (Filtered Back Projection) на линейные операторы, включающие в себя прямое и обратное преобразование Фурье, каждый из которых был имплементирован в обучающий фреймворк TensorFlow. Благодаря этому авторам удалось обучить восстанавливающую сеть в режиме end-to-end.

Нельзя не упомянуть и использование “кепстров” в комбинации с нейронными сетями для анализа сигналов. В работе [50] кепстральное разложение (наряду с другими данными) поступает на вход нейронной сети для анализа работы двигателей и для определения возможных проблем. В работе [51] описывается похожий метод для определения дефектов речи человека. А в работе [52] используется кепстральный анализ в совокупности с нейронной сетью для анализа ЭЭГ человека и определения эпилептических паттернов.

Другое интегральное преобразование, используемое в комбинации с нейронными сетями - вейвлет преобразование. Так, в работе [53] авторы используют дискретное вейвлет преобразование для сжатия данных и снижения уровня шума в них. Затем, поверх полученных признаков, они применяют полносвязную нейронную сеть. Подобная схема оказывается эффективной в задаче предсказания уровня грунтовых вод. Существенно более сложная модель предложена в работе [54]. Здесь предлагаемый метод используется для предсказания плотности движения. Его принципиальным отличием является рекуррентность: вейвлет преобразование принимает на вход не только исходные данные, но еще и результат работы нейронной сети на предыдущих.

Отдельно стоит отметить использование голосовательных схем на базе Хафа [55, 56, 57]. Такие работы делятся на два основных класса: 1) выбор ответа с помощью голосовательной схемы Хафа поверх выхода нейронной сети [58, 59], 2) обучение глубоких сетей поверх признаков, выбранных голосовательной схемой Хафа. Работ первого класса заметно больше, так как сама по себе идея использовать нейронную сеть для фильтрации изображения, вы-

деления признаков и построения описателей, поверх которых уже хорошо работает инженерный метод, естественна. Однако в работах второго типа приводится интересное наблюдение: так как после применения голосовательной схемы входные данные теряют большинство специфических признаков, то генерировать синтетические данные для обучения становится проще. Большинство таких работ относительно новые, что говорит об актуальности подобного рода исследований. В данных работах по большей части рассматриваются задачи, связанные с сегментацией трехмерных данных. Такие алгоритмы обрабатывают входную картинку с помощью глубокой нейронной сети, а уже поверх карт признаков строят голосовательные схемы и классификаторы, которые также могут быть нейросетевыми.

1.5 Выводы. Постановка задачи

Из описанных выше методов хорошо видна общая тенденция делать с помощью нейронной сети операции над изображениями, которые не переводят карты признаков в принципиально другой вид. Современные нейронные сети отлично справляются с задачами “улучшения” изображения в смысле, задаваемом функцией потерь, в случае, когда задача разрешима исходя из локальных признаков. Практика показывает, что даже когда это не так, тем не менее оптимизация функционала возможна. В то же время, интегральные преобразования по-прежнему фигурируют в алгоритмах как основные элементы, задающие поведение метода в целом. Они применяются там, где, наоборот, решение задачи не может быть построено, исходя лишь из локальных признаков. Выходами большого количества интегральных операторов от изображения является также изображение, но в совершенно другой системе координат. Наличие системы координат в таком изображении побуждает, как и в случае с исходным изображением, применить

сверточные нейронные сети. В качестве интегрального оператора в данной работе будет использоваться преобразование Хафа.

Несмотря на разностороннее применение нейронных сетей, вопрос использования преобразования Хафа внутри нейронных сетей плохо изучен. В работе показано, что добавление слоев преобразования Хафа способно существенным образом изменить свойства и поведение нейронной сети.

Так как преобразование Хафа является линейным оператором, то позволяет методу обратного распространения ошибки проходить сквозь него и обучать глубокие end-to-end модели. Другой факт, хорошо заметный из обзора, заключается в том, что подавляющее большинство методов содержат в себе лишь одно интегральное преобразование, в то время как существуют классические алгоритмы, включающие в себя два и более таких преобразований. Таким образом получается, что решение классических задач с помощью end-to-end нейронных сетей с использованием преобразования Хафа в виде слоев не исследовано. С другой стороны, подобный подход обещает решение ряда проблем: увеличение устойчивости к различным искажениям во входных данных по отношению к инженерным алгоритмам за счет нейросетевой обработки и упрощению по отношению к существующим нейросетевым подходам за счет добавления в сеть возможности работать с релевантными задаче признаками в явном виде.

Целью данной работы является исследование поведения нейронных сетей, включающих в свою архитектуру слои быстрого преобразования Хафа в явном виде в практических задачах компьютерного зрения. Для достижения поставленной цели необходимо решить следующие задачи:

1. Исследовать трансформации рецептивного поля при добавлении слоя быстрого преобразования Хафа. Это необходимо для мотивированного построения нейросетевых архитектур для решения конкретных задач.
2. Изучить способы и особенности распространения градиента через слои быстрого преобразования Хафа. Без такого

способа обучение нейронной сети в end-to-end режиме невозможно. Кроме этого, необходимо оценить вычислительную сложность и практическую применимость различных подходов.

3. Создать семейство нейросетевых архитектур для решения набора практических задач. Данные архитектуры должны быть объяснимы с точки зрения классического использования интегральных операторов.
4. Исследовать поведение сетей созданных архитектур при решении ими прикладных задач компьютерного зрения: поиск внутренней точки схода, поиск внешней точки схода, детекция выпуклых объектов. Показать, что нейросети разработанных архитектур способны показывать конкурентное качество при меньшем количестве обучаемых параметров и/или при требовании меньшего количества вычислительных мощностей.

Глава 2. БПХ как слой нейронной сети

2.1 О рецептивных полях

Прежде чем переходить к деталям построения нейронных сетей с использованием быстрого преобразования Хафа, остановимся подробнее на таком важном понятии, как рецептивное поле. Рецептивным полем нейрона называется влияющее на результирующее значение множество элементов его входа. Например, для сверточного фильтра его рецептивным полем является окно, по которому происходит вычисление. Если нейроны имеют пространственное упорядочение, можно говорить о форме и размере рецептивного поля (для сверточного слоя это прямоугольник, размер которого равен размеру окна фильтра). Когда говорится о рецептивном поле нейронной сети, имеется в виду рецептивное поле ее выходных нейронов относительно входного изображения. Именно из него определяется максимальный размер признаков, которые данная сеть способна анализировать.

Если размера рецептивного поля недостаточно для того, чтоб охватить какой-то объект целиком, то нейронная сеть не сможет использовать всю информацию об объекте независимо от количества обучаемых параметров, количества слоев и других ее характеристик. Этот факт делает задачу анализа формы и размера рецептивного поля сети неотъемлемой частью построения оптимальных архитектур.

Для сети с полносвязными слоями рецептивным полем является весь ее вход, а для полносверточной нейронной сети рецептивным полем является прямоугольник, размер которого можно вычислить, зная параметры используемых сверточных слоев. Схематично накопление размера рецептивного поля изображено на Рис. 2.1. Размеры такого прямоугольника для многослойной сети можно посчитать по формулам 2.1.

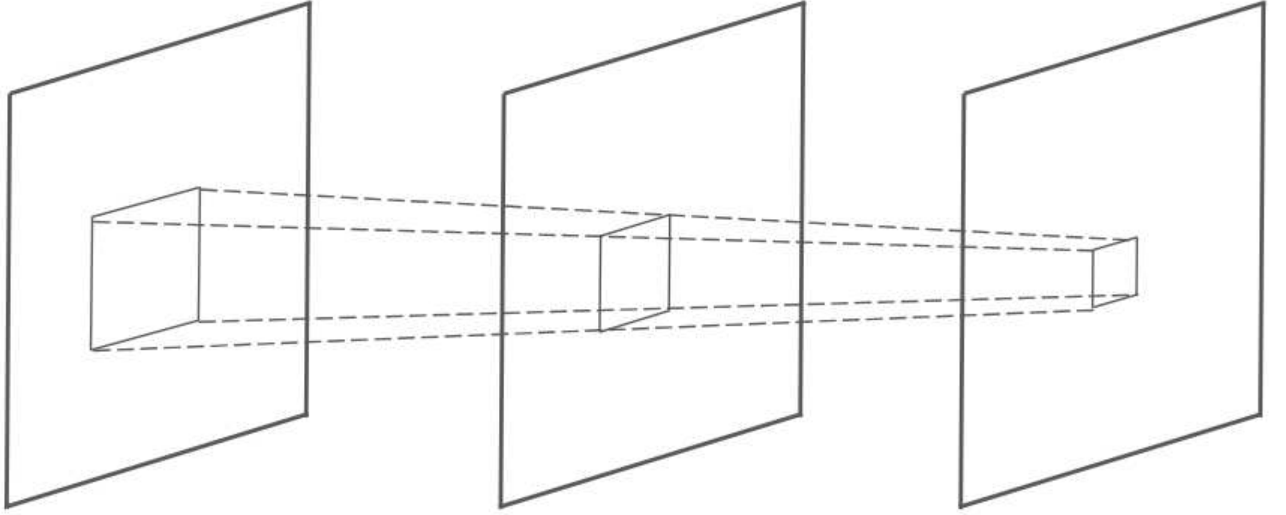


Рисунок 2.1 — Образование рецептивного окна в полносверточной сети.

$$\begin{cases} k_i = k_{i-1} * s_{i-1}/a_{i-1} \\ R_i = R_{i-1} + (f_i - 1) * k_{i-1}, \end{cases} \quad (2.1)$$

где $R_0 = 1$ и $k_0 = 1$. Таким образом получается, что стандартные полносверточные сети не всегда имеют способность решить задачу и, вообще говоря, их архитектура должна быть специальным образом подобрана. Можно добиться универсальности, сделав рецептивное поле очень большим, но, во-первых, это негативно отразится на качестве работы с маленькими объектами, а, во-вторых, приведет к тому, что архитектура окажется переусложнена и решение перестанет быть оптимальным.

Для некоторых случаев решить эту проблему можно путем добавления слоев интегральных преобразований. Рассмотрим простой пример: дана функция, которая везде равна нулю кроме двух точек x_0 и x_1 , в которых она равна единице, как показано на рисунке 2.2 слева. Необходимо построить такой сверточный слой (сверточную сеть), которая была бы способна для каждой точки x сказать, находится ли эта точка между x_0 и x_1 или нет. Для принятия такого решения блоку сверточных слоев необходимо видеть обе точки, иначе решить задачу нет возможности. Таким образом, сеть с рецептивным полем в три пикселя никогда не сможет решить поставленную задачу, если расстояние между x_0 и x_1 больше. Теперь

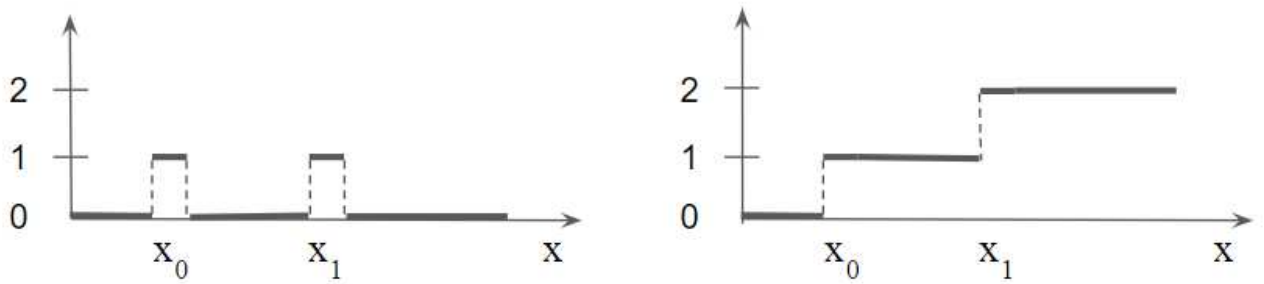


Рисунок 2.2 — Одномерная функция с двумя единичными точками (слева) и ее интеграл (справа).

проинтегрируем эту функцию, как показано на рисунке 2.2 справа. После такого преобразования поставленную задачу можно решить уже сетью с единичным рецептивным полем: если значение равно единице, значит точка лежит между x_0 и x_1 , иначе — снаружи. Таким образом простое интегральное преобразование позволяет серьезно снизить требование к нейронной сети при решении некоторых задач. Конечно, такого же эффекта можно было добиться полносвязным слоем, но это бы, во-первых, наложило ограничения на длину рассматриваемого отрезка (так как полносвязный слой требует всегда вход фиксированного размера) и, во-вторых, добавило бы огромное количество (n^2 , где n - длина отрезка) обучаемых параметров. Интегральное преобразование позволяет достигнуть цели, не вносит ограничений на длину и вообще не имеет обучаемых параметров.

В качестве такого интеграла при обработке двумерных карт признаков можно использовать БПХ. Рецептивное поле сети со слоями БПХ представляет из себя сложную структуру, в которую входят все пиксели изображения. Дело в том, что если для каждого элемента карты признаков до применения БПХ рецептивным полем является центрированный на нем прямоугольник из входной карты признаков, то после применения БПХ рецептивным полем каждого элемента выхода являются все пиксели соответствующей прямой на исходной карте признаков.

Если перед таким слоем есть сверточные слои, то получается, что рецептивное поле в каждой точке прямой это на самом деле

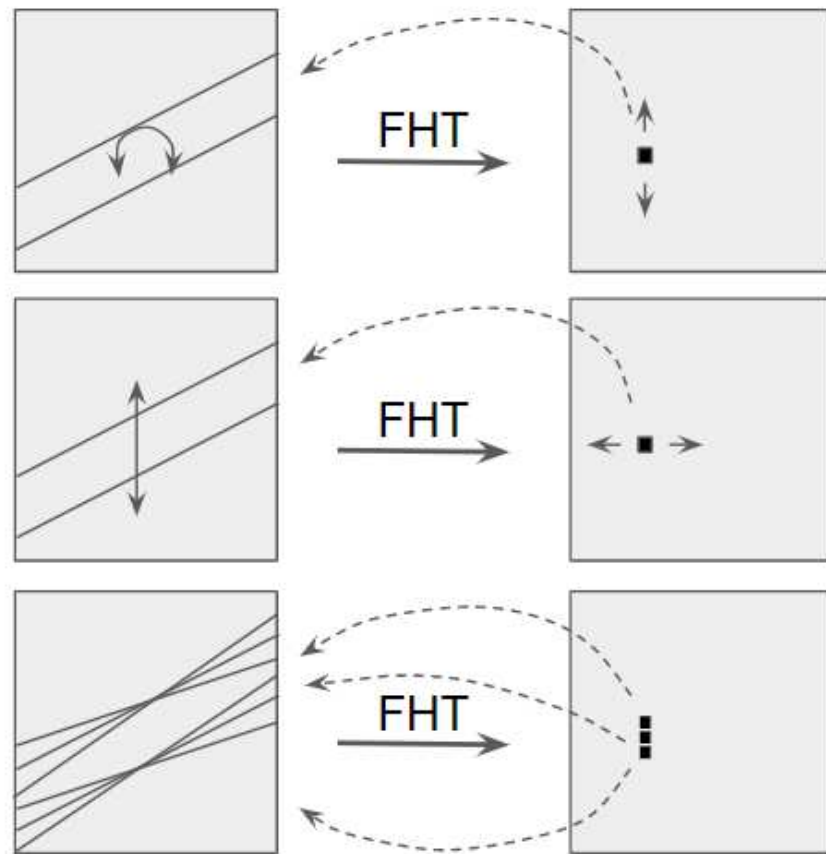


Рисунок 2.3 — Образование рецептивного окна при наличии одного слоя быстрого преобразования Хафа.

некоторый прямоугольник, размер которого зависит от конфигурации тех самых слоев. А сама прямая трансформируется в полосу с шириной, равной длине проекции прямоугольника на нормаль к прямой. В работе [32] доказано, что для повышения робастности поиска прямых на изображении с помощью интегральных преобразований необходимо использование оконных операций. Как можно видеть, нейронная сеть со слоем быстрого преобразования Хафа после сверточного блока способна как раз сделать именно такое преобразование. На Рис. 2.3 (сверху) схематически изображено рецептивное поле в виде полосы.

Отдельно рассмотрим изменение рецептивного поля при добавлении сверточного блока после слоя быстрого преобразования Хафа. Для этого необходимо рассмотреть не один пиксель выхода, а прямоугольный блок. Здесь размерности x и y будут иметь разный смысл, поэтому будем рассматривать их по отдельности. Координата x кодирует сдвиг прямой на исходном изображении и,

тем самым, расширение рецептивного поля по этой координате после БПХ слоя приводит к расширению полосы (т.е. рецептивного поля центрального пикселя), изображено на Рис. 2.3, снизу-слева. Иначе обстоит дело с расширением по y : это приводит не к сдвигам исходной полосы, а к вращениям. Таким образом, результирующее рецептивное поле конструкции “блок сверток – БПХ – блок сверток” – это набор полосок на изображении, центры которых проходят через общую точку, а количество определяется высотой рецептивного поля второго сверточного блока, изображено на Рис. 2.3, снизу-справа.

Если же после первого преобразования Хафа идет второе, то форма рецептивного поля становится еще сложнее и позволяет решать некоторые задачи, для которых требуется информация со всего входного изображения. Если сверточные слои отсутствуют, рецептивным полем выхода второго преобразования будет пучек прямых, пересекающихся в одной точке.

Такие формы рецептивных полей у сетей, использующих быстрое преобразование Хафа в виде слоев, позволяют сетям обретать свойства, заметно отличающиеся от свойств классических полно-сверточных сетей.

2.2 Прямое БПХ

Будем считать, что исходное изображение имеет размер (h, h) . Алгоритмически в преобразовании весь диапазон углов делится на четыре четверти, каждая из которых вычисляется независимо. В случае квадратного изображения это $H_1 : [-45, 0]$, $H_2 : [0, 45]$, $H_3 : [45, 90]$, $H_4 : [90, 135]$. Сразу оговоримся, что при вычислении суммы вдоль прямых будем считать, что изображение продолжается во все стороны нулями. Для четверти H_1 прямая задается точками x_0 и x_1 на верхней и нижней границах изображения, координата x_0 задает непосредственно сдвиг, а разность $x_1 - x_0$ задает

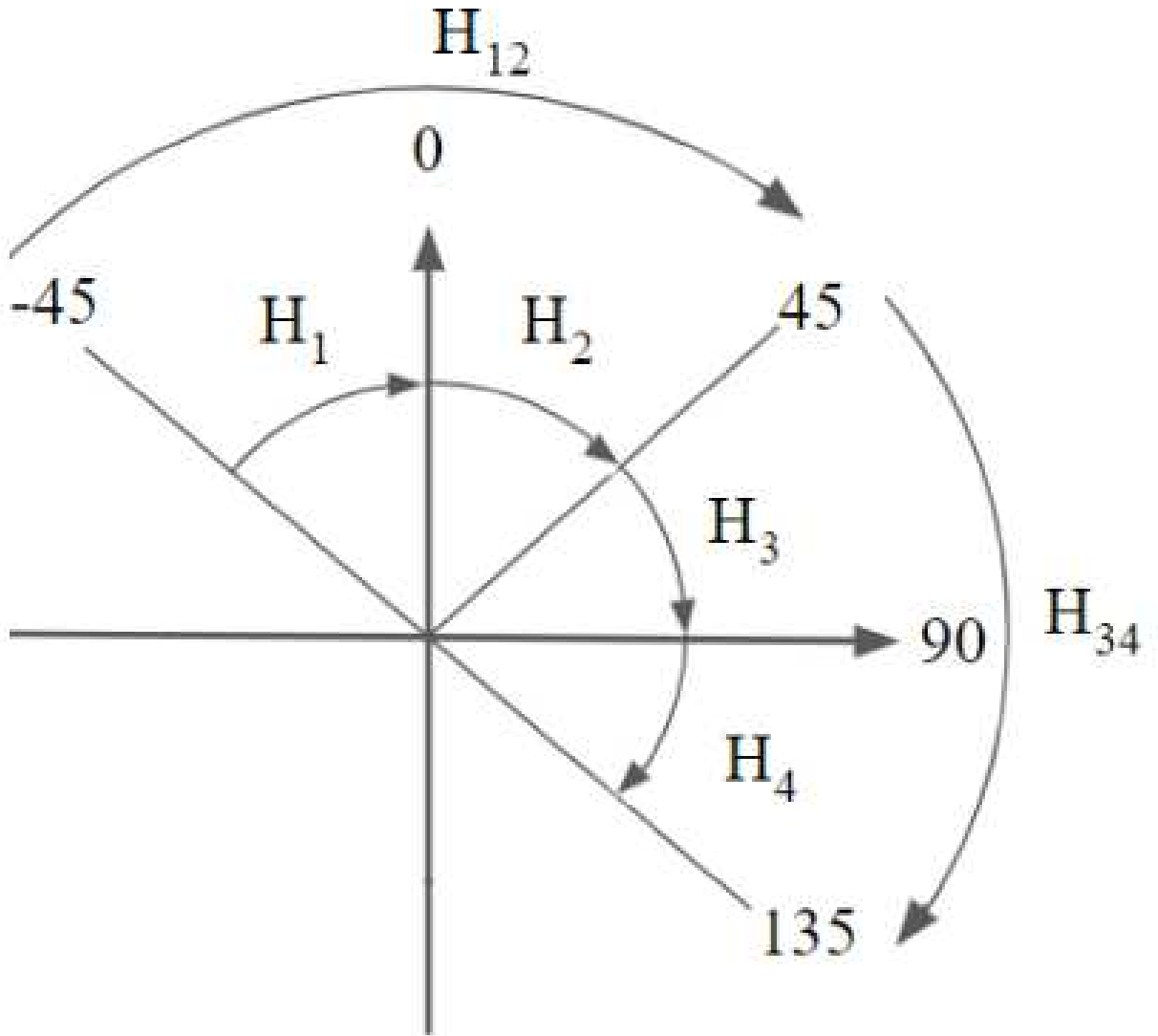


Рисунок 2.4 — Диапазоны углов для вычисления БПХ по четвертям.

смещение t . Так как для каждой из четвертей по построению получается h углов (для $t \in [0, h]$), то получается, что при объединении всех четвертей в одно изображение (обычно угол откладывается по вертикали) его размер по вертикали получается на три пикселя меньше, чем суммарный размер всех четвертей. В работе также будет использоваться H_{12} для преобразования по половине диапазона, состоящего из преимущественно вертикальных прямых, и H_{34} — для преимущественно горизонтальных соответственно. Диапазоны углов показаны на рисунке 2.4.

Для H_{12} , который соответствует преимущественно вертикальным линиям на изображении, можно посчитать координаты резуль-

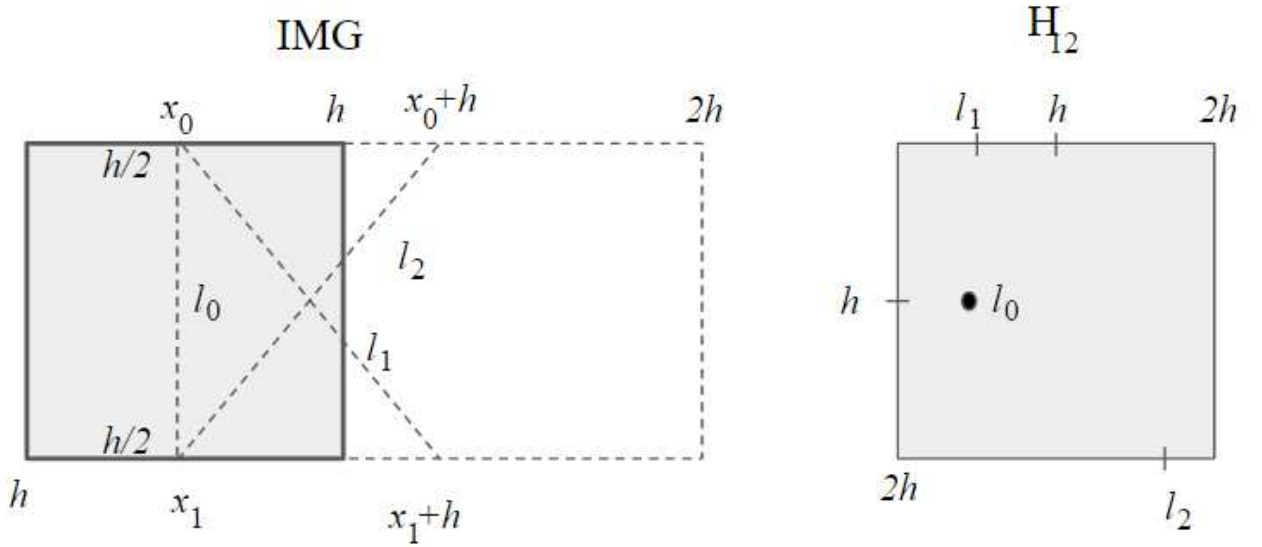


Рисунок 2.5 — Переход прямой в точку при быстром преобразовании Хафа.

тирующей точки (s, t) для прямолинейного сегмента $(x_0, 0) - (x_1, h - 1)$ на входном изображении с помощью уравнения 2.2:

$$\begin{cases} s = x_0 \\ t = h - (x_1 - x_0) \end{cases} \quad (2.2)$$

Для H_{34} сегмент прямой задается как $(0, y_0) - (h, y_1)$, а (s, t) могут быть вычислены с помощью уравнения 2.3:

$$\begin{cases} s = y_0 \\ t = h - (y_0 - y_1) \end{cases} \quad (2.3)$$

Рассмотрим преобразование H_{12} подробнее на примере изображения размером $h \times h$ как показано на рисунке 2.5. Точка x_0 находится на верхней стороне, а точка x_1 — на нижней. Таким образом, согласно формуле 2.2, вертикальные прямые на исходном изображении перейдут в точки, лежащие на высоте h на результирующем изображении (линия l_0 на исходном изображении), прямые под углом -45° перейдут в точки на высоте 0 на результирующем изображении (линия l_1), а прямые под углом 45° перейдут в точки на высоте $2h$ (линия l_2) на результирующем изображении.

На рисунке 2.6 слева показано исходное изображение. На нем можно видеть точки (сверху темнее, снизу ярче). Каждая из этих точек переходит в ломаную на центральной и правой картинках

с соответствующей интенсивностью. В центре показан результат БПХ. Если посмотреть на результирующее изображение, то можно видеть, что сама картинка, хоть и является цельной, но переходит через левый край изображений и продолжается справа в нижней и верхней частях. С точки зрения обработки такого изображения как вектора это не играет роли. Однако в нашей работе мы будем проводить оконные операции над таким изображением. Для того, чтоб можно было применять стандартные методы без дополнительных сложностей, будем делать “центрирование”, т.е. циклический сдвиг строк на величину, зависящую от угла (вертикальной координаты). Такой сдвиг позволяет центрировать полученное изображение и использовать стандартные сверточные слои для его корректной обработки. Такой сдвиг вычисляется по формуле 2.4, а результат его применения показан на рисунке справа:

$$\begin{cases} d_v(t) = h - t/2 \\ d_h(t) = t/2 \end{cases} \quad (2.4)$$

В таком случае формулы 2.2 и 2.3 для преимущественно вертикальных и преимущественно горизонтальных прямых соответственно модифицируются добавлением сдвига:

$$\begin{cases} s = x_0 + d_v(t) \\ t = h - (x_1 - x_0) \end{cases} \quad (2.5)$$

$$\begin{cases} s = y_0 + d_h(t) \\ t = w - (y_0 - y_1) \end{cases} \quad (2.6)$$

Отметим, что при сдвиге пикселизованных прямых может наблюдаться эффект “расчески”, продемонстрированный на рисунке 2.7. Изучение этой проблемы и степени ее влияния на качество работы нейронных сетей находится за пределами данной диссертации и будет предметом последующих исследований.



Рисунок 2.6 — Исходное изображение (слева), БПХ без сдвига (в центре), БПХ со сдвигом (справа).

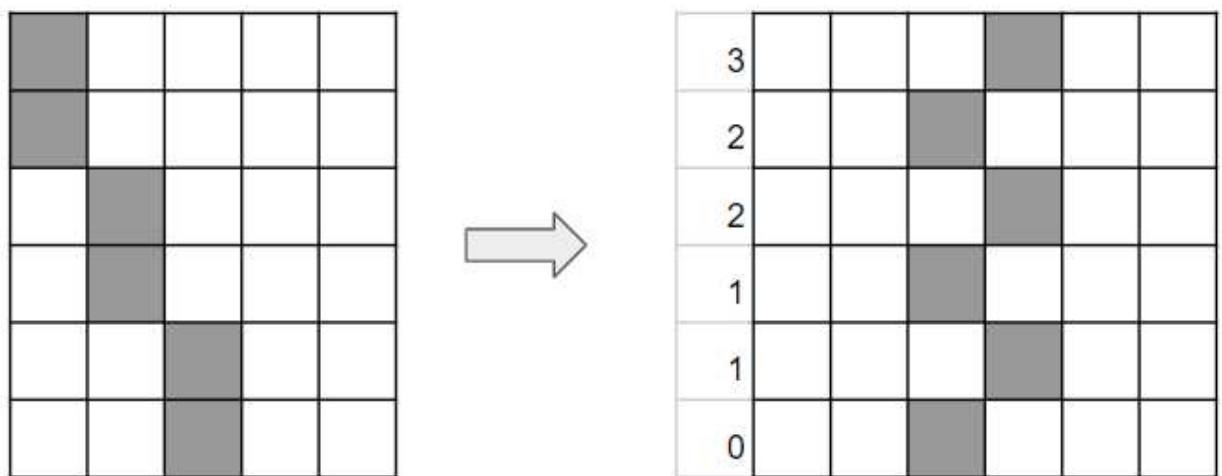


Рисунок 2.7 — Образование “расчески” при сдвиге пикселизированной прямой.

2.3 Диадические паттерны прямых в БПХ

Перед тем как перейти к рассмотрению транспонированного БПХ необходимо подробно остановиться на том, как строятся паттерны прямых, по которым происходит суммирование в быстром преобразовании Хафа. В отличие от классической параметризации в преобразовании Радона, где каждая прямая задается ее углом α и расстоянием от центра ρ , в БПХ каждая прямая задается ее начальной точкой на стороне изображения s и сдвигом конечной точки на противоположной стороне (вертикальные стороны для горизонтальных прямых и горизонтальные для вертикальных) относительно начальной t . В рассматриваемом преобразовании используется так называемый диадический паттерн прямых, который является рекуррентным, что, в свою очередь, позволяет повторно использовать частичные суммы и улучшить асимптотику алгоритма. Именно это и делает данное преобразование “быстрым”. Между точками в пространстве (ρ, φ) и (s, t) имеется взаимнооднозначная связь, которую можно выразить в виде уравнений 2.7 [60]:

$$\begin{cases} \tan(\varphi) = -(\frac{h}{t})^p \\ \rho = \frac{sh}{\sqrt{t^2 + h^2}}, \end{cases} \quad (2.7)$$

где h - сторона изображения, а p принимает значение 1 или -1 в зависимости от рассматриваемой четверти преобразования.

В работе [30] дается аналитическое определение таких паттернов. Для простоты будем рассматривать только случай квадратного изображения со стороной $h = 2^p$. Рассмотрим двоичное представление значений x и t , т.е. набор нулей и единиц такие, что:

$$t = \sum_{r=0}^{p-1} t_r 2^r; x = \sum_{i=0}^{p-1} x_i 2^i \quad (2.8)$$

Тогда можно определить матрицу $H(x, t)$, которая бы задавала смещение каждого пикселя в паттерне длины 2^p :

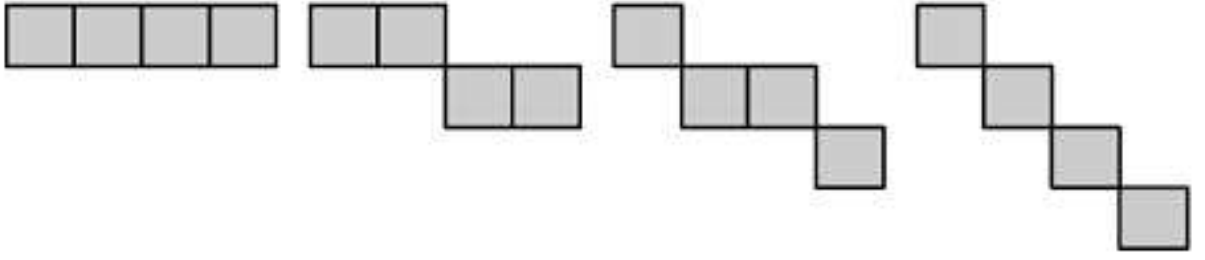


Рисунок 2.8 — Все возможные паттерны для длины 4.

$$H(x, t) = \sum_{r=0}^{p-1} t_r \left[\frac{2^r x}{2^p - 1} \right], \quad (2.9)$$

где x — номер пикселя в паттерне, а t — смещение последнего пикселя относительно первого.

Примеры паттернов для $p = 2$ изображены на рисунке 2.8. По этому рисунку можно видеть, что, например, $H(0,0) = 0$, $H(3,3) = 3$, $H(1,2) = 1$ и т.д. Подробное описание способа построения диадических паттернов и некоторые его свойства подробно описаны в работе [61].

В случае, если изображение не квадратное и/или его размер не представим как 2^p , где p — целое, описанные методы построения прямой в представленном виде применить нельзя. Тем не менее, существуют аппроксимации и адаптации этих алгоритм, исследование которых лежит за пределами данной работы и не требуется для анализа работоспособности нейросетевых архитектур, которые будут рассмотрены.

2.4 Распространение градиента через БПХ

Для того, чтобы нейронные сети можно было учить методами градиентного спуска, необходимо уметь распространять градиент через все слои (за исключением некоторых особых случаев, например, через первый слой в сети распространение градиента не

требуется). В качестве нового слоя в данной работе рассматривается слой, выполняющий быстрого преобразования Хафа для карт признаков. Как было показано во введении, каждый элемент выхода такой операции является линейной комбинацией всех элементов входа с весом 0 или 1, т. е. представляет собой сумму некоторых из них. В таком виде преобразование является линейным оператором и может быть записано в виде матрицы. Если ввести правило преобразования карты признаков в вектор, то можно рассматривать быстрое преобразование Хафа, как умножение входного вектора на такую матрицу.

При реализации будущих слоев будет использован быстрый алгоритм, но для доказательства некоторых свойств прямого и транспонированного операторов БПХ будет использоваться матричная запись оператора. Это позволит использовать линейную алгебру и работу с матрицами в явном виде.

Так как БПХ является линейным оператором, то при выполнении нейронной сети такой слой можно рассматривать как операцию умножения входного вектора признаков на матрицу весов (которые для данного оператора predetermined и равны либо 0, либо 1). Для обучения предыдущих слоев необходимо на них передавать градиент через такой слой при обратном распространении ошибки. Эта операция достигается путем умножения вектора входящего с последующего слоя градиента на транспонированную матрицу оператора.

Еще одним важным аспектом является то, что транспонированный оператор можно использовать не только для распространения градиента, но и в качестве операции слоя. Тогда, при распространении градиента, будет использоваться транспонированный от него, т.е. прямой оператор. Это свойство будет использовано для расширения семейства рассматриваемых нейросетевых архитектур, построенных с использованием прямого и транспонированного БПХ.

2.5 Транспонированное БПХ

В данном разделе будет рассмотрен оператор транспонированного Быстрого преобразования Хафа и доказано, что применение такого оператора идентично следующей последовательности операций: отражение входного изображения относительно вертикальной оси, применение прямого оператора быстрого преобразования Хафа, отражение результата относительно вертикальной оси. Таким образом, транспонированный оператор может быть посчитан не просто с такой же асимптотической сложностью, а теми же самыми функциями, что и прямой [62]. Благодаря этому можно с одинаковой эффективностью использовать слои как прямого, так и транспонированного БПХ, а возможность переиспользования реализаций функций хоть и не обещает серьезной практической значимости, тем не менее, представляет удобство для написания и отладки программы, реализующей этот функционал.

Для начала заметим, что если утверждение верно для какой-то из четвертей, то оно верно для всех, так как один диапазон углов в другой переводится путем отражения и поворота исходного изображения. В дальнейшем доказательстве будем рассматривать преобразование в одной четверти.

Для доказательства данного утверждения необходимо рассмотреть прямой и транспонированный операторы БПХ в матричном виде. Для того, чтоб представить карту признаков в виде вектора, пронумеруем ее элементы таким образом, что сначала номер получат все элементы первой строки слева направо, затем второй и т.д.: $I_{i*n+j} := \tilde{I}_{ij}$, $i, j = \overline{1, n}$, $\tilde{I} \in X$. Тогда множество всех карт признаков размера (n, n) может быть рассмотрено как векторное пространство X , в котором имеется $n * n$ базисных векторов. Операции суммы, умножения на число и скалярного произведения для таких векторов определены.

Значение каждого элемента результирующей карты признаков является линейной комбинацией значений элементов входной

0	0	0	0
0	0	1	1
0	1	1	2
0	1	2	3

Рисунок 2.9 — Матрица смещения паттернов

карты признаков, а результирующая карта признаков, следовательно, действие операции БПХ может быть рассмотрено как действие линейного оператора $\mathcal{A} : X \rightarrow X$. Обозначим A матрицу $\mathcal{A}$. Будем также называть матрицу A матрицей быстрого преобразования Хафа. Матрица Грамма для X единичная, следовательно, $\mathcal{A}^*$ (транспонирование от $\mathcal{A}$) это A^T .

Рассмотрим матрицу $H(x, t)$, введенную в разделе 2.3 и задающую тангенциальный сдвиг пикселя относительно стартовой точки в зависимости от длины паттерна и номера пикселя в нем. Например, для картинке $4 * 4$ такая матрица будет иметь размер $4 * 4$, потому что все рассматриваемые дискретные прямые будут иметь длину 4 и количество возможных сдвигов между первым и последним элементом тоже 4. По оси x откладывается номер элемента в паттерне, а по оси t – его сдвиг относительно первого. Получается, что для смещения 0 паттерн имеет просто прямолинейный вид, а для, например, смещения 2 нулевой пиксель не смещается, следующие два смещаются на одну позицию и последний еще на одну. Пример такой матрицы для длины 4 представлена на Рис. 2.9.

Лемма 1. $H(x, t) = H(t, x)$ для фиксированного n . Другими словами, матрица, составленная из сдвигов паттернов симметрична.

Доказательство этой леммы было приведено в работе [63], но оно основано на другом определении паттерна, а в работе [61] используется такое же определение паттерна, но утверждение используется без доказательства.

Доказательство. ► Заменим вектор x его разложением по степеням двойки и затем заменим остаток внутри операции округления суммой остатков:

$$H(x, t) = \sum_{r=0}^{p-1} t_r \left[\frac{2^r x}{2^p - 1} \right] = \sum_{r=0}^{p-1} t_r \left[\sum_{i=0}^{p-1} \frac{2^r x_i 2^i}{2^p - 1} \right]. \quad (2.10)$$

Заметим, что $\forall y \in \mathbb{R} \quad \frac{2^y}{2^p - 1} = 2^{y-p} + \frac{2^{y-p}}{2^p - 1}$, и, следовательно $\forall r = \overline{0, p-1}$:

$$\left\{ \left\{ \frac{x_i 2^r 2^i}{2^p - 1} \right\}, i = \overline{0, p-1} \right\} = \left\{ \left\{ \frac{x_i 2^i}{2^p - 1} \right\}, i = \overline{0, p-1} \right\} =: S \quad (2.11)$$

Можно заметить два важных свойства S : только один элемент в S может быть больше, чем $\frac{1}{2} \cdot \frac{x_{p-1} 2^{p-1}}{2^p - 1}$, а еще сумма элементов в S не может быть больше, чем 1:

$$\sum_{i=0}^{p-1} \frac{x_i 2^i}{2^p - 1} \leq \sum_{i=0}^{p-1} \frac{2^i}{2^p - 1} = \frac{1}{2^p - 1} \left(\sum_{i=0}^{p-1} 2^i \right) = 1. \quad (2.12)$$

Перепишем 2.10 в следующем виде:

$$\left[\sum_{i=0}^{p-1} \frac{x_i 2^r 2^i}{2^p - 1} \right] = \left[\sum_{i=0}^{p-1} \left(\left\lfloor \frac{x_i 2^r 2^i}{2^p - 1} \right\rfloor + \left\{ \frac{x_i 2^r 2^i}{2^p - 1} \right\} \right) \right] = \sum_{i=0}^{p-1} \left\lfloor \frac{x_i 2^r 2^i}{2^p - 1} \right\rfloor + \left[\sum_{i=0}^{p-1} \left\{ \frac{x_i 2^r 2^i}{2^p - 1} \right\} \right]. \quad (2.13)$$

Можно заметить, что если $\{y\} < \frac{1}{2}$, тогда $\lfloor y \rfloor = [y]$. Из этих двух свойств S можно заключить следующее:

$$\sum_{i=0}^{p-1} \left\lfloor \frac{x_i 2^r 2^i}{2^p - 1} \right\rfloor = \sum_{i=0}^{p-1} \left[\frac{x_i 2^r 2^i}{2^p - 1} \right] - x_{p-1-r} \quad (2.14)$$

и

$$\sum_{i=0}^{p-1} \left\{ \frac{x_i 2^r 2^i}{2^p - 1} \right\} = x_{p-1-r} \quad (2.15)$$

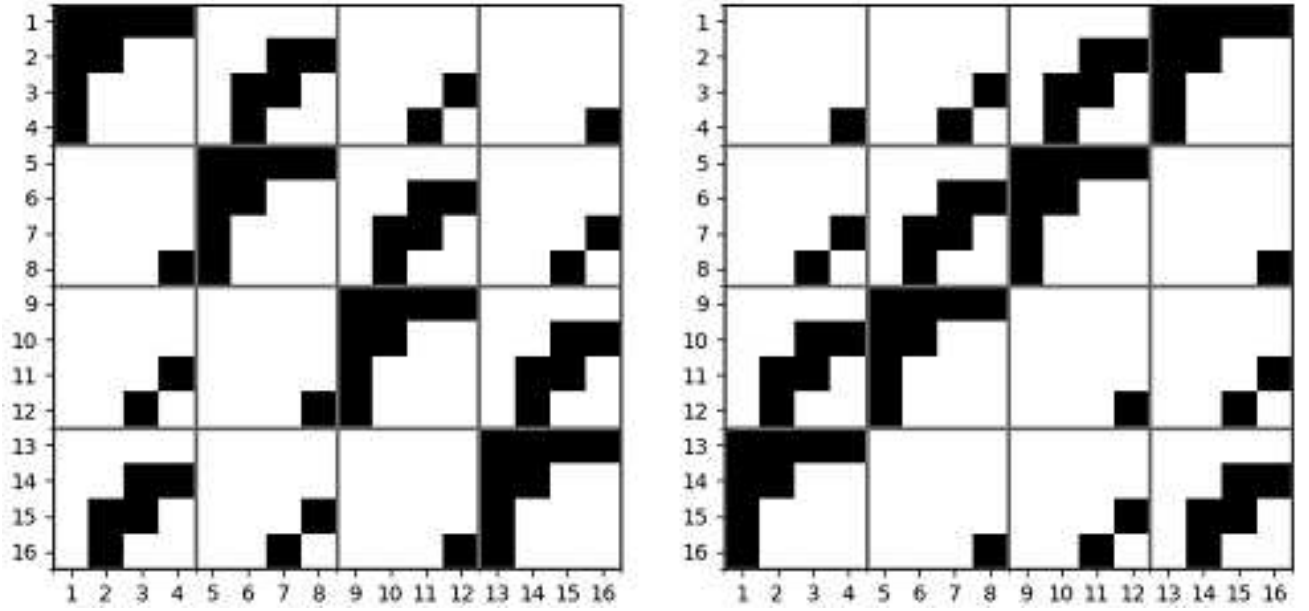


Рисунок 2.10 — Матрица БПХ, исходная слева, преобразованная справа.

В результате, комбинируя 2.10, 2.13, 2.14 и 2.15:

$$\begin{aligned}
 H(x, t) &\stackrel{2.10}{=} \sum_{r=0}^{p-1} t_r \left[\sum_{i=0}^{p-1} \frac{2^r x_i 2^i}{2^{p-1}} \right] \stackrel{2.13}{=} \\
 &\stackrel{2.13}{=} \sum_{r=0}^{p-1} t_r \left(\sum_{i=0}^{p-1} \left\lfloor \frac{x_i 2^r 2^i}{2^{p-1}} \right\rfloor + \left[\sum_{i=0}^{p-1} \left\{ \frac{x_i 2^r 2^i}{2^{p-1}} \right\} \right] \right) \stackrel{2.14, 2.15}{=} \\
 &\stackrel{2.14, 2.15}{=} \sum_{r=0}^{p-1} t_r \left(\sum_{i=0}^{p-1} \left\lfloor \frac{x_i 2^r 2^i}{2^{p-1}} \right\rfloor \right) = \\
 &= \sum_{r=0}^{p-1} \sum_{i=0}^{p-1} t_r x_i \left\lfloor \frac{2^r 2^i}{2^{p-1}} \right\rfloor = H(t, x)
 \end{aligned} \tag{2.16}$$



Рассмотрим матрицу быстрого преобразования Хафа как блочную матрицу $\mathbf{B}$, в которой каждый блок имеет размер (n, n) как показано на изображении 2.10 слева. Блоки $\mathbf{B}_{i,j}$ будем обозначать как (i, j) блоки.

$$A = \left(\begin{array}{c|c|c|c} \mathbf{B}_{1,1} & \mathbf{B}_{1,2} & \cdots & \mathbf{B}_{1,n} \\ \hline \mathbf{B}_{2,1} & \mathbf{B}_{2,2} & \cdots & \mathbf{B}_{2,n} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \mathbf{B}_{n,1} & \mathbf{B}_{n,2} & \cdots & \mathbf{B}_{n,n} \end{array} \right)$$

Лемма 2. Блоки матрицы симметричны.

Доказательство. ► $A_{m,q}$ – проекция входного пикселя q на выходной пиксель m , следовательно $\mathbf{B}_{i,j}$ – это проекция строки входных пикселей $\{(j-1)*n+k, k=\overline{1,n}\}$ на строку выходных пикселей $\{(i-1)*n+k, k=\overline{1,n}\}$. Тогда

$$(\mathbf{B}_{i,j})_{k,l} = \begin{cases} 1 & , H(k,l) = n - i + j \pmod{n} \\ 0 & \text{в противном случае} \end{cases}$$

где H – ранее определенная функция смещения. Согласно лемме 1 $H(k,l) = H(l,k)$, а значит $\mathbf{B}_{i,j}$ симметрична. ◀

Лемма 3. $\mathbf{B}_{i,j} = \mathbf{B}_{i+k-1 \pmod{n}+1, j+k-1 \pmod{n}+1}$ для всех $i, j = \overline{1,n}$, $k = \overline{0, n-1}$.

►

◻

Вращение входного изображения циклически относительно вертикальной оси приведет к соответствующему вращению выходного изображения. Столбцы матрицы быстрого преобразования Хафа это выходные изображения из X с базисными векторами и столбцами x_i и x_{i+n} , следовательно, вращение относительно вертикальной оси на n пикселей приводит к тому, что $A_{ij} = A_{i+n, j+n}$. Таким образом в матрице $\mathbf{B}$ элементы, находящиеся на одной диагонали равны.

◀

Обозначим $n \times n$ блоков матрицы AC как $\tilde{\mathbf{B}}_{ij}$ (как показано на изображении 2.10 справа), а блоки матрицы CAC как $\hat{\mathbf{B}}_{ij}$.

Лемма 4. AC симметрична.

Доказательство. ► Правое умножение матрицы A на матрицу C переставляет столбцы матрицы A . Более того, из определения матрицы $\mathcal{C}$ можно заключить, что такое умножение переставляет столбцы и матрицы $\mathbf{B}$, т.е. не изменяет расположение столбцов матрицы A внутри их групп $\{(i-1)*n+k, k=\overline{1,n}\}$, $i=\overline{1,n}$. $\mathbf{B}_{i,j} = \tilde{\mathbf{B}}_{n+1-i,j}$, $\tilde{\mathbf{B}}_{j,n+1-i} = \mathbf{B}_{n+1-j, n+1-i} = \mathbf{B}_{(i,j)+(1,1) \times (n+1-i-j)}$. Из леммы 3 следует, что $\mathbf{B}_{i,j} = \mathbf{B}_{(i,j)+(1,1) \times (n+1-i-j)}$, следовательно $\tilde{\mathbf{B}}$

симметрична. Из леммы 2 следует, что $\mathbf{V}_{i,j}$ симметричны и тогда AC тоже симметрична. ◀ □

Теорема 1. *Обозначим $\mathcal{C}$ оператор отражения изображения: $\mathcal{C}e_{i*n+j} = e_{(n+1-i)*n+j}$, где $i, j = \overline{1, n}$. Тогда $\mathcal{C}\mathcal{A}\mathcal{C} = \mathcal{A}^*$ (или $CAC = A^T$, где C матрица из $\mathcal{C}$).*

Заметим, что $C = C^{-1}$, а тогда A^T это тоже матрица из $\mathcal{A}$ в базисе $\langle \mathcal{C}x_i \rangle$. Для доказательства используем леммы 1, 2, 3, 4.

Доказательство. ▶ $\mathbf{V}_{i,j} = \tilde{\mathbf{V}}_{n+1-i,j}$. Из леммы 4 следует, что матрица AC симметрична, а значит $\tilde{\mathbf{V}}_{n+1-i,j} = \tilde{\mathbf{V}}_{j,n+1-i}$. Левое умножение матрицы AC на матрицу C переставляет строки матриц AC и $\tilde{\mathbf{V}}$. Тогда $\mathbf{V}_{i,j} = \tilde{\mathbf{V}}_{j,n+1-i} = \hat{\mathbf{V}}_{j,n+1-(n+1-i)} = \hat{\mathbf{V}}_{j,i}$. Из леммы 2 следует, что $\mathbf{V}_{i,j}$ симметричны, а значит $CAC = A^T$. ◀ □

Таким образом, прямое и транспонированное быстрое преобразование Хафа можно вычислять не только с одинаковой асимптотической сложностью, но и использовать для этого общую реализацию. Это позволит, во-первых, строить эффективные архитектуры нейронных сетей с использованием транспонированного преобразования в виде слоя, во-вторых, быстрее учить нейронные сети, так как при распространении градиента от слоя прямого преобразования необходимо вычислять транспонированное для него.

2.6 Инвариантность к позиции в Хафе и Радоне

Необходимо упомянуть еще одно свойство пространства (s, t) , отличающее его от пространства (ρ, φ) , где по оси x отложен сдвиг, а по оси y отложен угол. Параметр t , используемый для задания угла в быстром преобразовании Хафа хоть и однозначно связан с углом, но представляет не угол, а сдвиг между началом и концом

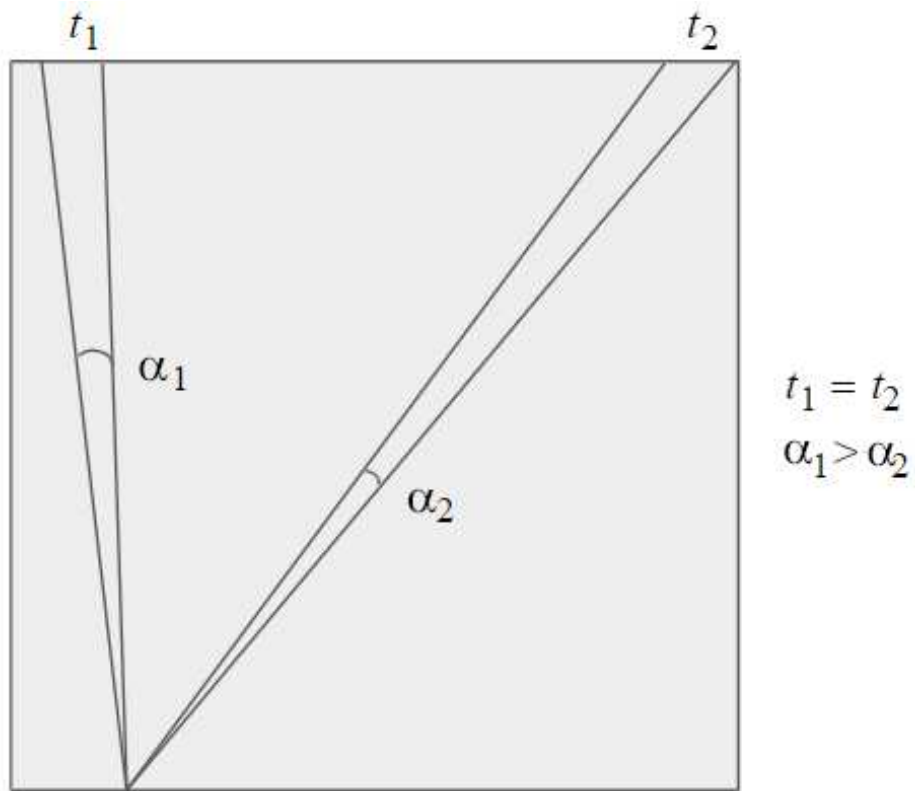


Рисунок 2.11 — Приращение угла нелинейно зависит от сдвига.

паттерна, соответствующего задаваемой прямой. В итоге получается, что приращение угла при изменении t не постоянно и зависит от текущего значения t :

$$d\alpha = f(s), \quad (2.17)$$

Этот эффект проиллюстрирован на Рис. 2.11, из которого наглядно ясно, почему так происходит. Здесь $d\alpha_1 > d\alpha_2$. Это, в свою очередь, приводит к тому, что на Хаф-образе появляются неоднородности: разные точки в зависимости от координаты s будут иметь различные рецептивные поля.

Такой особенности лишен результат преобразования Радона, так как угол при его вычислении изменяется с постоянной скоростью. В большом количестве инженерных алгоритмов эта особенность не играет особой роли, но в рассматриваемом случае это, вероятно, не так. Дело в том, что обучаемые сверточные фильтры имеют окна заранее заданного размера и хорошо известно, что их слабым местом является отсутствие устойчивости к масштабированию. Здесь имеется похожая проблема и, следовательно, разумно

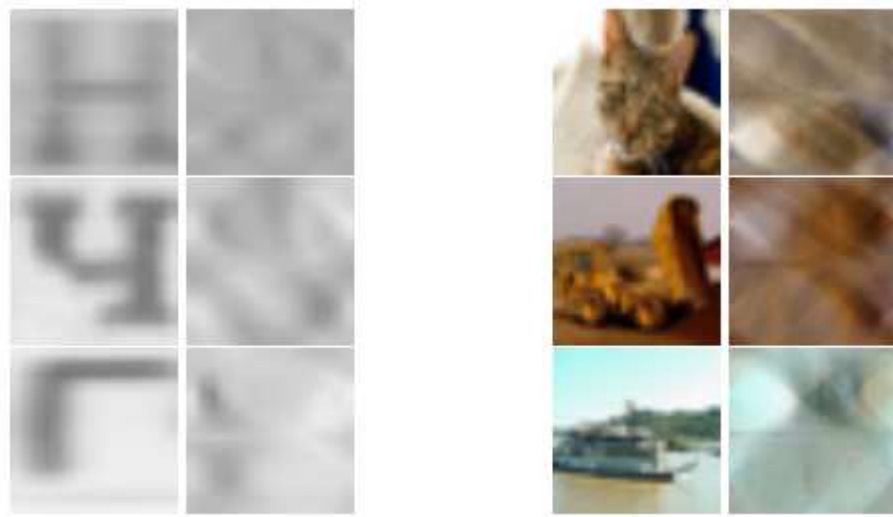


Рисунок 2.12 — Примеры изображений и результатов БПХ для них.

предположить, что обучение сверточных фильтров может столкнуться с определенными трудностями. Однако это исследование также выходит за рамки данной диссертации и представляет собой тему для дальнейших исследований.

2.7 Сохранение информации

Прежде чем перейти к созданию и обучению нейронных сетей с использованием слоев прямого и транспонированного БПХ, остановимся еще на одном вопросе: годится ли Хаф-образ изображения для использования в качестве входа нейронной сети, так как применение БПХ к исходному изображению его существенно трансформирует. Изображения трансформируются настолько, что человек (как минимум неподготовленный) не в состоянии их узнавать. На Рис. 2.12 показаны примеры изображений и их преобразований для букв (слева) и различных объектов (справа).

Прежде чем изучать особенности поведения нейронных сетей со слоем быстрого преобразования Хафа внутри, необходимо убедиться, что вся специфическая информация, необходимая для классификации, сохраняется. Для этого была проведена серия

экспериментов [64, 65], целью которых было установить, что Хаф-образ изображения пригоден для использования в качестве входа для слоев нейронной сети.

В этих экспериментах использовалась сверточная нейронная сеть, состоящая из двух ветвей (в каждой по сверточному слою и по слою пулинга). Их выходы конкатенировались и подавались на вход полносвязному слою. Все фильтры в обоих ветвях имели фиксированный размер 5×5 , а их количество изменялось. Для одной ветви количество фильтров равнялось 16, 32, 64 или 128, а для второй – 4, 8, 16 или 32 соответственно. Количество обучаемых параметров в полносвязном слое изменялось, так как размер его входа в каждом эксперименте был разным, потому что зависел от количества фильтров.

В первом наборе экспериментов использовалось исходное изображение в качестве входа для обеих ветвей сети. Во втором наборе экспериментов в качестве входа одной из ветвей сети использовался Хаф-образ изображения. Важно заметить, что в таком случае паддинг будет иметь несколько другой смысл, так как локальные зависимости между пикселями в исходном пространстве и в Хаф-пространстве различаются. На вторую ветвь сети все также подавалось исходное изображение.

Результаты этих экспериментов собраны в Таблице 1 для наборов данных CIFAR-10 и для текстовых строк. Для подавления эффектов, связанных со случайной инициализацией при обучении все эксперименты были проведены 10 раз, а их результаты усреднены. По результатам видно, что замена части входов на Хаф-образ изображения позволяет улучшить качество при всех рассмотренных количествах фильтров. Из результатов также следует, что перевод изображения в Хаф-образ позволяет сверточной сети получить из изображения больше информации при сохранении общего числа фильтров, что делает идею использования таких изображений в качестве входов сверточных слоев перспективной. Особенно это может быть важно в приложениях, где критически важна про-

изводительность (например, в [66]) и в случаях, когда используются специальные методы ускорения работы [67, 68, 69, 70].

Таблица 1 — Точность различных нейросетевых архитектур при наличии и отсутствии результатов преобразования Хафа во входе второй ветви на наборах данных CIFAR-10 и строках из русского паспорта. N_{source} — количество фильтров в ветви для исходного изображения, N_{hough} — количество фильтров в ветви для преобразования Хафа.

N	N_{source}	N_{hough}	CIFAR-10		Russian passport	
			Error rate, without Hough	Error rate, with Hough	Error rate without Hough	Error rate with Hough
1	16	4	47.5%	43.4%	7.2%	6.9%
2	32	8	43.8%	42.7%	6.5%	6.3%
3	64	16	40.3%	37.9%	6.2%	6.2%
4	128	32	35.7%	34.1%	6.1%	6.0%

Из таблицы 1 видно, что использование результата преобразования Хафа в качестве входа на часть фильтров вместо оригинального изображения позволяет нейронной сети получить больше информации из входа и, следовательно, улучшить качество работы. И хотя улучшение качества незначительно, оно присутствует во всех случаях. Кроме этого, ни изображения с набора данных CIFAR-10, ни изображения букв по своей сути не нуждаются в Преобразовании Хафа, так как они маленькие по размеру. Можно заметить, что влияние тем больше, чем меньше фильтров в сети в целом, что тоже не противоречит гипотезе: чем сеть сложнее, тем больше информации она способна получить из изображения сама по себе и тем меньше добавится от входов, преобразованных Хафом.

Таким образом, даже в случае, когда причины для этого не очевидны, использование Преобразования Хафа в комбинации со сверточными нейронными сетями позволяет сети получить релевантную информацию.

2.8 Результаты главы 2

В данной главе подробно рассмотрено быстрое преобразование Хафа и параметризация (s, t) его выходного пространства и показано, что при использовании этого преобразования в виде слоя нейронной сети для обучения необходимо вычислять транспонированное быстрое преобразование Хафа. В главе приводится доказательство того, что транспонированное преобразование также может быть вычислено быстро, т.е. со сложностью $O(n^2 * \log(n))$, а не $O(n^3)$, как в наивной реализации. Проведенный анализ модификации рецептивного поля выходных нейронов в нейронной сети со слоем быстрого преобразования Хафа показал, добавление преобразования Хафа существенно его увеличивает.

Иными словами, в главе 2 было:

1. Показано, что добавление слоя быстрого преобразования Хафа между сверточными слоями нейронной сети кардинальным образом меняет размер и структуру рецептивного поля нейронов, что, в теории, способно наделить результирующую нейронную сеть новыми свойствами.
2. Показана возможность построения быстрого транспонированного оператора для быстрого преобразования Хафа. Это, во-первых, позволяет эффективно распространять градиент через такой слой при обучении, а, во-вторых, позволяет переводить признаки из пространства (s, t) обратно в пространство (x, y) , что будет использоваться при составлении архитектур нейронных сетей в дальнейшем.
3. Приведено доказательство того, что алгоритм транспонированного оператора для быстрого преобразования Хафа имеет такую же асимптотическую сложность, как и прямое быстрое преобразование Хафа, т.е. $O(n^2 * \log(n))$.

Глава 3. Архитектуры сетей с БПХ и способы их применения

В данной главе будут рассмотрены архитектуры нейронных сетей, использующие слои прямого и транспонированного БПХ. Как было показано в предыдущих главах, инженерные алгоритмы, использующие интегральные преобразования, встречаются повсеместно, а задачей данной работы является встраивание быстрого преобразования Хафа в нейронные сети и исследование свойств и возможностей таких сетей. Основываясь на представленных в главе 2 свойствах быстрого преобразования Хафа будет создан слой нейронной сети, совершающий это преобразование над картами признаков. Благодаря этому слою, нейронная сеть сможет работать по образу соответствующего алгоритма.

Рассмотрим три принципиально разных варианта использования такого нейросетевого алгоритма, которые определяются особенностями постановки задачи.

В первом варианте будет осуществляться поиск внешних точек схода. Такая задача возникает, например, при анализе трехмерных сцен или фотографии документов в произвольной ориентации в пространстве. В подавляющем большинстве случаев такие изображения содержат точки схода, которые лежат за их пределами (вплоть до бесконечности). Их нахождение позволяет оценить позиции объектов интереса в кадре и, следовательно, критически важно для дальнейшего анализа изображения. Так, например, при обработке изображений с документом точки схода образуются его сторонами и строками текста в нем. Если такие точки найти, то можно далее провести “ректификацию” изображения [71] для дальнейшей обработки. Задача поиска внешних точек схода при анализе изображений городских сцен, например, возникает при передвижении беспилотного средства, которому необходимо знать свою ориентацию и ориентироваться в пространстве.

Классический алгоритм поиска несобственной точки схода базируется на последовательном применении двух БПХ. Первое

преобразование дает максимумы в позициях, соответствующих прямым линиям на исходном изображении, а второе дает максимумы в позициях, соответствующих точкам схода. Необходимо отметить, что между ними еще необходимо некоторое нелинейное преобразование, иначе в силу инварианта Радона второе преобразование не будет иметь смысла. На этом и будет базироваться первая архитектура, названная Хафнет. Важной деталью тут является то, что необходим алгоритм, способный корректно обрабатывать точки, лежащие в бесконечности, с чем легко справится предлагаемая нейросетевая архитектура. В общих чертах такая сеть состоит из последовательности сверточных слоев и слоев прямого БПХ. Из-за специфики БПХ такая сеть состоит из нескольких ветвей (будет подробно описано далее в пункте 3.2), выходом каждой из которых является двумерная карта признаков, каждая точка которой соответствует определенной точке в координатах исходного изображения, т.е. может быть в том числе и за пределами фактической картинки. Обучается сеть таким образом, чтобы в точках, соответствующих позициям точек схода в плоскости входного изображения, значения были максимальными.

Во втором варианте точка схода внутренняя. Решение такой задачи часто необходимо при анализе, например, дорожных сцен, снятых с видеорегистратора. Найденная точка схода от дороги (рельсов, стен тоннеля) может позволить беспилотному средству двигаться в заданном направлении и/или контролировать свою позицию относительно краев дороги. Такие изображения, содержащие точку, соответствующую пересечению продолжений краев дороги, часто в позиции самой точки схода часто не содержат какую-либо релевантную информацию. Это делает решение данной задачи локальными методами невозможным. Для решения такой задачи также можно использовать алгоритм, базирующийся на двух последовательных применениях БПХ, но так как нет необходимости обрабатывать точки, лежащие в бесконечности, в архитектуре сети это можно учесть и сделать ее значительно более удобной в использовании. Для решения такой задачи предлагает-

ся полносверточная архитектура Хафэнкодер, базирующаяся на применении прямого и транспонированного БПХ к изображению и, тем самым, возвращающая ответ в исходных координатах. По аналогии с предыдущей архитектурой в этой архитектуре также предполагается ответ сети в виде двумерной карты признаков. Главное отличие заключается в том, что эта карта признаков в тех же самых координатах и такого же размера как и исходное изображение и, следовательно, искомые точки находятся путем банального взятия координат пикселей, значения в которых представляют собой локальные максимумы.

Дополнительно стоит заметить, что выходы таких сетей позволяют искать не один максимум на результирующей карте признаков, а, вообще говоря, сколько угодно и, тем самым, получать набор альтернатив с их условной уверенностью, т.е. сети обеих архитектур пригодны для поиска множественных точек схода в рамках решаемой задачи.

Третий вариант – семантическая сегментация изображений, т.е. классификация каждого пикселя по признаку принадлежности тому или иному объекту. Наиболее качественные решения подобной задачи показывают U-Net-подобные сети, но мы продемонстрируем, что использование HoughEncoder сети здесь позволяет существенно улучшить различные качества итогового решения, такие как количество обучаемых параметров, качество работы. В данном случае выходом сети является двумерная многоканальная карта признаков в каждом пикселе которой находится распределение псевдовероятностей принадлежности данного пикселя к соответствующему классу.

3.1 Полносвязная сеть со слоями БПХ

Для решения задачи локализации точки схода внутри изображения классический алгоритм также содержит в себе двойное

преобразование Хафа. В отличие от поиска внешней точки схода, второе преобразование Хафа делается по преимущественно горизонтальным прямым. Далее, для локализации объекта с заданной точностью можно разбить результирующее изображение на решетку с соответствующим количеством ячеек. Если говорить про построение нейросетевого алгоритма, то для этого случая можно в конец нейронной сети добавить полносвязные слои, где количество выходов последнего полносвязного слоя должно совпадать с количеством ячеек решетки. В таком случае эту сеть можно учить как классификатор: она должна классифицировать входные изображения по номеру ячейки, в которую попадает точка схода.

Данный подход является самой простой адаптацией использования нейронной сети с БПХ к задаче поиска внутренних точек схода и имеет ряд серьезных недостатков [72]. Наличие полносвязных слоев приводит к огромному количеству обучаемых параметров, что приводит как к увеличению самой сети, так и к требованию большего количества обучающих данных. Более того, такая сеть будет иметь склонность переобучаться на позициях, которых представлено больше, а в силу специфики некоторых задач дисбаланс в данных вполне возможен. Так, например, при поиске точки схода на дороге она будет чаще ближе к середине изображения. Дополнительные проблемы возникают из-за наличия ячеек: чем их больше, тем больше обучаемых параметров в сети и выше шанс на ошибку, а чем их меньше, тем ниже точность правильного ответа.

Тем не менее, такая архитектура будет рассмотрена в экспериментах в качестве стартовой точки для некоторых задач и позволит произвести корректное сравнение с уже опубликованными результатами.

3.2 Архитектура Хафнет

Среди задач поиска точек схода поиск внешней точки схода является важным частым случаем. Такая задача часто возникает при анализе геометрических трехмерных сцен по двумерным изображениям (фотографиям), например, при восстановлении геометрической схемы в трехмерном пространстве в городах или при поиске ориентации документа для его последующего выправления. Т.е. для нахождения воображаемых точек, в которых бы пересеклись прямые, на которых лежат противоположные стороны документа. Особенностью таких точек является то, что они очень часто находятся далеко за пределами изображения (вплоть до бесконечности) и, следовательно, простыми методами обработки изображений их не найти. Для решения этой задачи необходим алгоритм, способный решать эту проблему.

Хотя архитектура с полносвязным слоем на конце, как описано в предыдущей секции, и способна решать задачу, она, как уже было сказано ранее, обладает рядом существенных разнородных недостатков. Дополнительно к ним, такая архитектура не подразумевает никакой зависимости между ответами в соседних пикселях, что противоречит природе задачи.

В качестве развития архитектуры можно отказаться от полносвязных слоев. Архитектура, приведенная в Таблице 2, может быть использована для задачи поиска несобственных точек схода [73].

Представленная архитектура нейронной сети основана на использовании слоя быстрого преобразования Хафа, который переводит карты признаков из (x, y) координат в (s, t) . Так как в рассматриваемом случае этот слой является линейным оператором, то обратное распространение градиента через него может быть выполнено путем транспонирования матрицы оператора. Важным фактом является то, что добавление такого слоя не увеличивает количество обучаемых параметров в сети, а лишь позволяет ей работать с признаками в другом пространстве.

Таблица 2 — Архитектура Хафнет

Слои			
#	Тип	Параметры	Функция активации
1	сверт	12 фильтры 5×5 , шаг 1×1 , без расширения	<i>relu</i>
2	сверт	12 фильтры 5×5 , шаг 2×2 , без расширения	<i>relu</i>
3	сверт	12 фильтры 5×5 , шаг 1×1 , без расширения	<i>relu</i>
4	БПХ	H_{12} для вертикальных, H_{34} для горизонтальных	-
5	сверт	12 фильтры 3×9 , шаг 1×1 , без расширения	<i>relu</i>
6	сверт	12 фильтры 3×5 , шаг 1×1 , без расширения	<i>relu</i>
7	сверт	12 фильтры 3×9 , шаг 1×1 , без расширения	<i>relu</i>
8	сверт	12 фильтры 3×5 , шаг 1×1 , без расширения	<i>relu</i>
9	БПХ	H_{34}	-
10	сверт	16 фильтры 5×5 , шаг 3×3 , без расширения	<i>relu</i>
11	сверт	16 фильтры 5×5 , шаг 3×3 , без расширения	<i>relu</i>
12	сверт	1 фильтры 5×5 , шаг 1×1 , без расширения	$1 - rbf$

Используя формулы 2.5, 2.6, 2.4, можно вычислить зависимость между координатами точек и координатами сегментов прямых между входными и результирующими картами признаков. Каждая точка на входной карте признаков переходит в различные сегменты для H_{12} и H_{34} . Если точка на исходной карте признаков задана координатами (x, y) , то координаты результирующих сегментов могут быть вычислены с помощью следующих формул:

$$\begin{cases} s = x + \frac{(\alpha - h)y}{h} + d_v(\alpha) \\ \alpha = [0; 2h - 1] \end{cases} \quad (3.1)$$

$$\begin{cases} s = y - \frac{(\alpha - w)x}{w} + d_h(\alpha) \\ \alpha = [0; 2w - 1] \end{cases} \quad (3.2)$$

Используя уравнения 2.6, 2.5, 2.4, 3.1, 3.2 возможно вывести уравнения для зависимости координат точки в исходном пространстве и в пространстве двойного преобразования Хафа $H_m(H_n)$:

$$\begin{cases} s' = \frac{3h}{2} + \frac{4hx - w(2y + h)}{2(h - 2y)} \\ \alpha' = \frac{(2y + h)(w + h)}{2y - h} \end{cases} \quad (3.3)$$

$$\begin{cases} s' = \frac{3w}{2} + \frac{4wy + h(2x - 3w)}{2(2x - w)} \\ \alpha' = \frac{(2x - 3w)(w + h)}{2x - w} \end{cases} \quad (3.4)$$

Уравнение 3.3 соответствует преобразованию $H_{34}(H_{12})$, а уравнение 3.4 – $H_{34}(H_{34})$.

При известных s' и α' можно вычислить координаты точки на исходном изображении (координатная система которого подверглась двойному преобразованию Хафа) с помощью следующих уравнений для $H_{34}(H_{12})$ и $H_{34}(H_{34})$ соответственно:

$$\begin{cases} x = \frac{\alpha'w + (3h - 2s')(w + h)}{2(\alpha' - (w + h))} \\ y = \frac{h(\alpha' + w + h)}{2(\alpha' - (w + h))} \end{cases} \quad (3.5)$$

$$\begin{cases} x = \frac{w(\alpha' - 3(w + h))}{2(\alpha' - (w + h))} \\ y = \frac{\alpha'h + (3w - 2s')(w + h)}{2(\alpha' - (w + h))} \end{cases} \quad (3.6)$$

3.3 Архитектура Хафэнкодер

Архитектура, описанная в секции 3.2, является универсальной и способна находить в том числе и собственные точки схода, но довольно сложна. В случае, когда известно, что точка схода находится внутри изображения, можно сделать проще, так как нет необходимости параметризовать всю бесконечную плоскость. Замена второго преобразования Хафа на транспонированное приведет к тому, что сеть станет полносверточным энкодером, выдающим ответ в тех же координатах, что и входное изображение, но сохранит способность оперировать интегральными признаками. Архитектуру, построенному по такому принципу, будем называть Хафэнкодером [74]. Кроме этого, такая архитектура способна решать задачу классификации пикселей входного изображения и, следовательно, может быть применима для решения задачи семантической сегментации. В общих чертах архитектура такой сети представлена на рисунке 3.1.

Использованный в работе пример такой архитектуры подробно описан в таблице 3. Функция активации `softsign` использовалась в силу ее простоты, везде ненулевой производной и ограниченности выходного значения. Шаг 0.5×0.5 означает, что данный слой является сверточным слоем специального вида, применение которого приводит к кратному увеличению размера карт признаков. Такие слои называют “апконволюцией”, “деконволюцией” [75] или даже “транспонированной сверткой” [76]. Есть некоторые различия в том, каким конкретно образом они увеличивают размер изображения, но суть у них одна: сделать изображение большего размера и, по возможности, подавить артефакт “шахматной доски”.

В данной конкретной архитектуре пятый слой выполняет преобразование системы координат на картах признаков согласно уравнениям 2.5 и 2.6, а слой 10 производит транспонированное преобразование Хафа и, тем самым, возвращает координатную систему обратно из (s, t) в (x, y) . Слои 6, 7, 8, 9, в итоге, работают

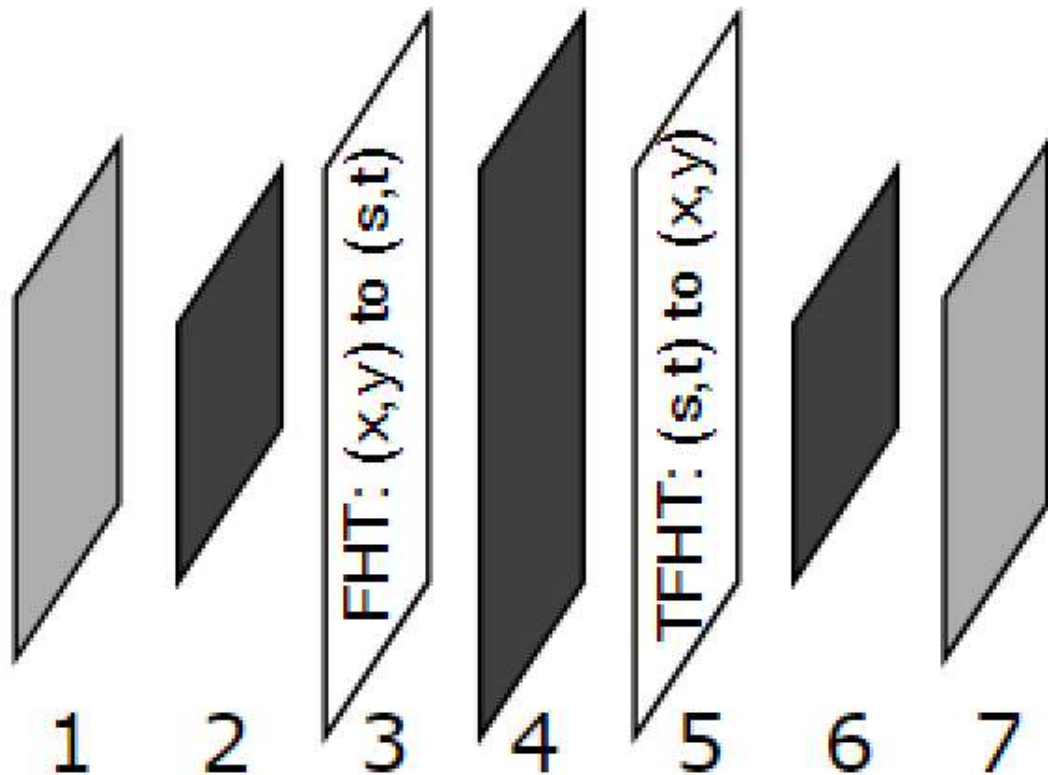


Рисунок 3.1 — Архитектура Хафэнкодера в общих чертах. Блоки 1 и 7 - вход и выход соответственно. Блок 2 - сверточные слои, работающие с исходным изображением. Блоки 3 и 5 слои прямого и транспонированного БПХ. Блок 4 - сверточные слои, работающие в координатах (s, t) . Блок 6 - сверточный блок, работающий в координатах (x, y) , но уже с глобальными признаками.

на картах признаков в координатах (s, t) и вычисляют сложные нелинейные признаки по различным линейчатым структурным элементам на входном изображении. Несмотря на то, что сами по себе эти слои сверточные и работают только с локальными признаками, благодаря сделанной трансформации системы координат они по сути работают уже с интегральными признаками вдоль различных прямых линий на исходном изображении. Кроме этого, существуют работы, показывающие, что оконные операции необходимы в том случае, когда данные зашумлены и содержат в себе ложные примеры [32]. Во всех сверточных слоях использовалось расширение карт признаков (паддинг) в режиме отражения для сохранения размеров и избежания появления ложных границ на краях. В итоге, при обработке изображения нейронной сетью такой архитектуры размер выходных карт признаков соответствует размеру входных,

Таблица 3 — Архитектура нейронной сети Хафэнкодер

#	Тип	Параметры	Функция активации
1	сверт	4 фильтры 3×3 , шаг 1×1	softsign
2	сверт	8 фильтры 3×3 , шаг 2×2	softsign
3	сверт	16 фильтры 3×3 , шаг 2×2	softsign
4	сверт	16 фильтры 3×3 , шаг 1×1	softsign
5	БПХ	—	—
6	сверт	16 фильтры 3×3 , шаг 1×1	softsign
7	сверт	16 фильтры 3×3 , шаг 1×1	softsign
8	сверт	16 фильтры 3×3 , шаг 1×1	softsign
9	сверт	16 фильтры 3×3 , шаг 1×1	softsign
10	ТБПХ	—	—
11	сверт	8 фильтры 3×3 , шаг 0.5×0.5	softsign
12	сверт	4 фильтры 3×3 , шаг 0.5×0.5	softsign
13	сверт	4 фильтры 3×3 , шаг 1×1	softsign
14	сверт	2 фильтры 3×3 , шаг 1×1	softmax

а сама сеть, следовательно, может быть использована как автоэнкодер. В то же время, благодаря присутствию слоев прямого и транспонированного преобразований Хафа, результат в каждой точке рассчитывается исходя не из некоторого прямоугольного рецептивного окна (как для стандартных полносверточных сетей), а исходя из линейчатых признаков, собранных под различными углами и с различными сдвигами. В результате, рецептивное поле сети сильно отличается от того, которое набирается имеющимися сверточными слоями.

Благодаря такой архитектуре, признаки в среднем блоке слоев Хафэнкодера выглядят совершенно не так, как в стандартных полносверточных архитектурах. Если в сетях таких архитектур, как UNet, входное изображение трансформируется в ответ относительно монотонно, в Хафэнкодере все иначе: есть три блока слоев, разделенных слоями быстрого преобразования Хафа, которые выполняют принципиально разные роли. На рисунке 3.2 показана сравнительная трансформация признаков в предложен-

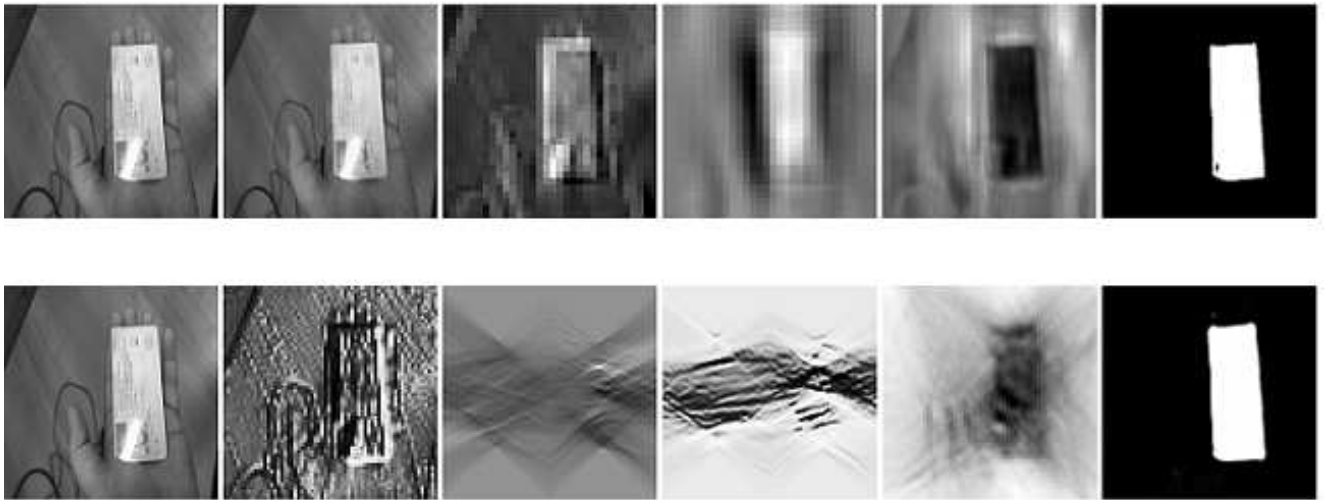


Рисунок 3.2 — Трансформация признаков в Хафэнкодере и в UNet.

ной сети и в сети архитектуры UNet. Снизу находятся карты признаков, взятые из слоев Хафэнкодера: входное изображение, карты признаков перед слоем прямого преобразования, карты признаков после слоя прямого преобразования, карты признаков перед слоем транспонированного преобразования, карты признаков после слоя транспонированного преобразования и результат. Сверху показана трансформация признаков в классической полносверточной сети. Можно видеть, что в Хафэнкодере происходит переход в (s, t) координаты, а потом восстанавливается исходная структура. Таким образом, эта сеть способна использовать глобальные признаки для построения ответа в каждой локальной точке.

3.4 Исследование возможностей Хафэнкодера

В параграфе 2.1 приводилось рассуждение, согласно которому использование интеграла от одномерной функции позволяет решить проблему ограниченности рецептивного поля для некоторых задач. С точки зрения рецептивного поля получается, что в каждой точке интеграла рецептивное поле – интервал от начала до взятой точки. В случае БПХ каждая точка это сумма значений во всех пикселях, лежащих на некоторой прямой. Таким образом, ре-

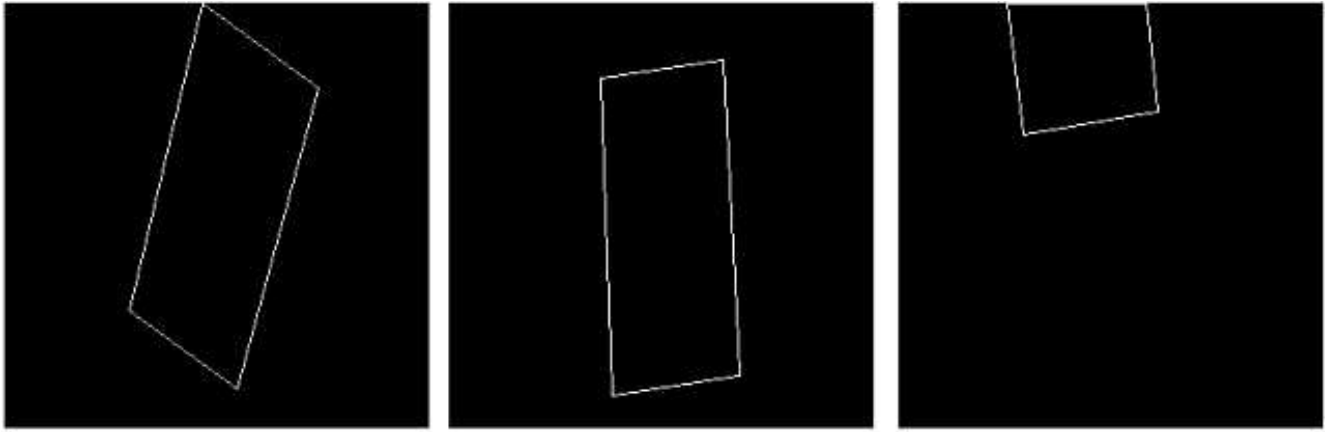


Рисунок 3.3 — Примеры контуров для экспериментов.

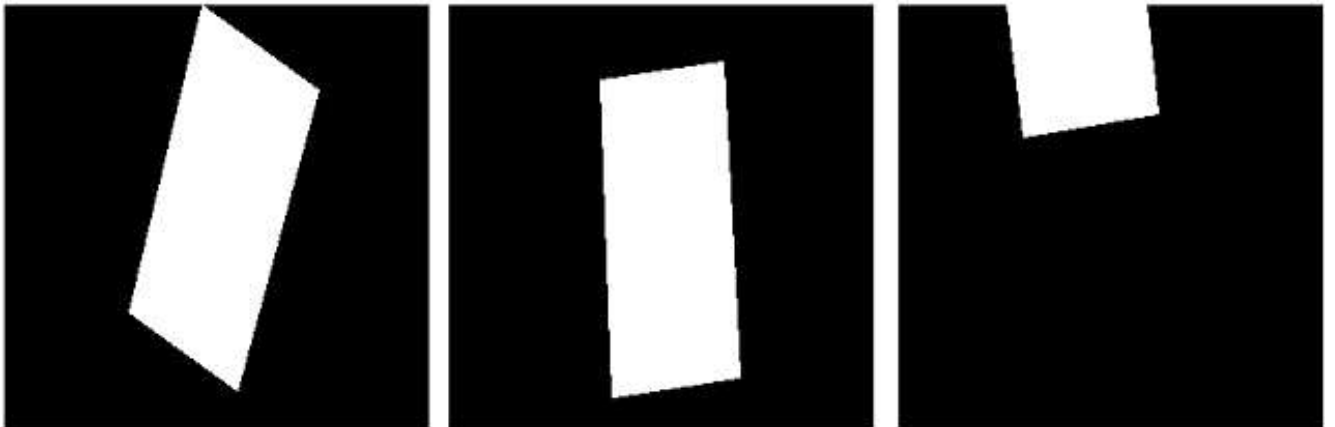


Рисунок 3.4 — Примеры идеалов для контуров с изображения 3.3.

цептивное поле каждой точки после БПХ – это интервал от начала до конца. Если бы рассматриваемая область не была ограничена, можно было бы говорить, что для интеграла рецептивное поле – луч, а для преобразования Хафа – прямая.

Рассмотрим двумерный случай: пусть дан контур выпуклого четырехугольника, как показано на рисунке 3.3. Поставим перед нейронной сетью следующую задачу: необходимо закрасить внутреннюю часть четырехугольника. Для этого к каждой из такой картинок сделаем идеал в виде такого же четырехугольника, но только закрашенного. Примеры идеалов для тех же четырехугольников продемонстрированы на рисунке 3.4.

Очевидно, что для принятия решения, находится ли каждая взятая точка внутри четырехугольника или снаружи, для выходного нейрона необходимо иметь рецептивное поле достаточного размера. Ведь если точка черная, а в ее рецептивное поле не попали

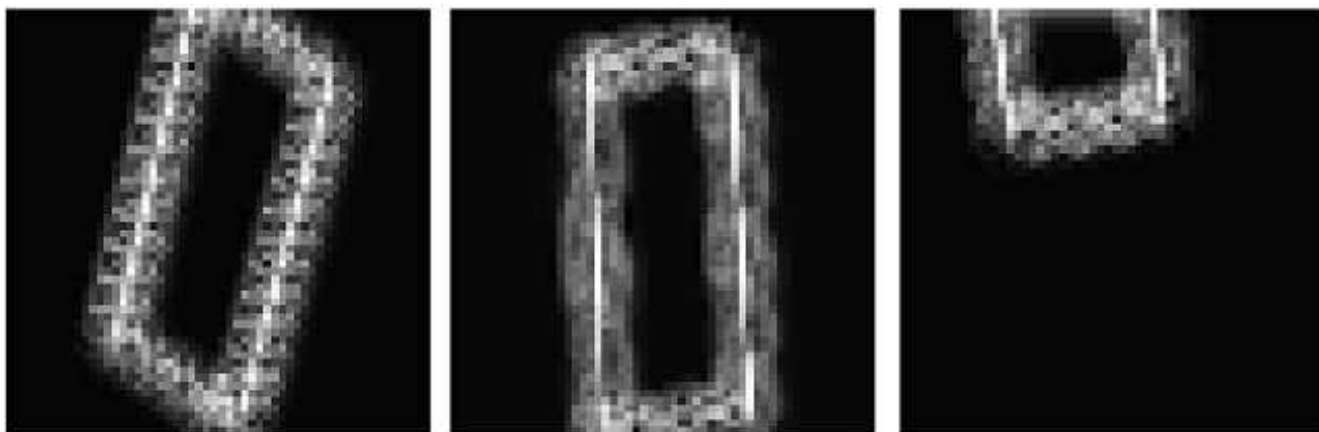


Рисунок 3.5 — Ответы полносверточной сети для изображений 3.3.

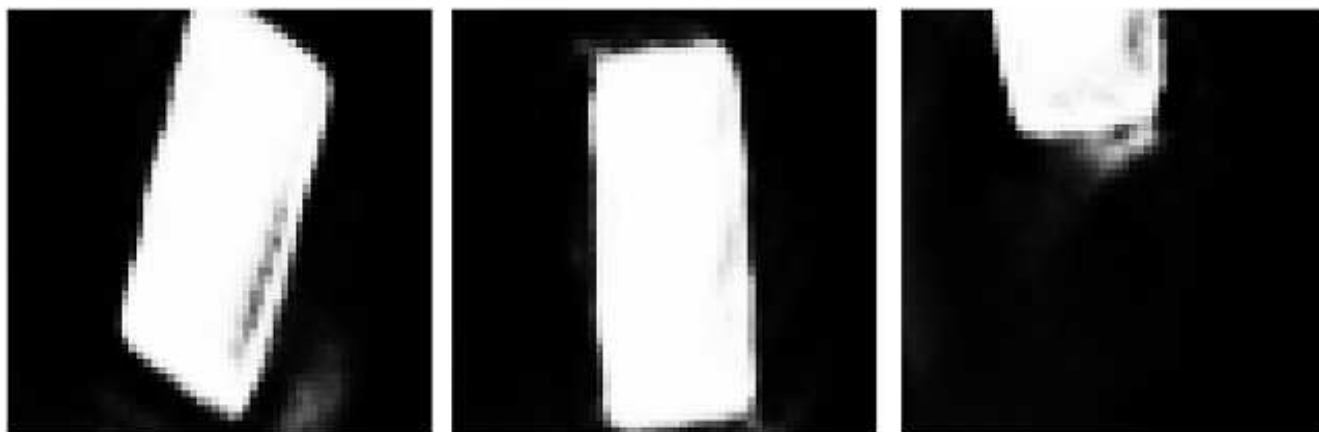


Рисунок 3.6 — Ответы Хафэнкодера для изображений 3.3.

никакие границы, то просто нет возможности сделать вывод о ее местонахождении. Для иллюстрации этого была обучена обычная полносверточная сеть с небольшим рецептивным полем, которая, ожидаемо, не смогла решить задачу. На выходе такой сети были получены изображения, как показано на рисунке 3.5.

Теперь добавим в данную нейронную сеть слой прямого БПХ (для трансформации рецептивного поля) после первого сверточного слоя и слой транспонированного БПХ (для того, чтобы получить выход в координатах (x, y) , а не (s, t)) после четвертого сверточного слоя. Несмотря на то, что количество обучаемых параметров не изменилось, рецептивное поле выходных нейронов стало принципиально иным. Теперь на каждую точку выхода может повлиять любая точка входа, и задача по закрашиванию четырехугольника для нейронной сети становится решаемой. Примеры ответов такой сети показаны на рисунке 3.6.

3.5 Влияние слоя преобразования Хафа на требуемое количество операций

Отдельно рассмотрим вычислительную сложность предлагаемых слоев в сравнении с другими слоями нейронной сети. Количество требуемых операций - $O(cs^2 \log s)$, где c – количество каналов изображения, s – линейный размер изображения. Для сверточного слоя требуется порядка $O(cs^2 f^2 m)$ операций, где f – линейный размер фильтра, m – количество фильтров. Соотношение между вычислительными сложностями сверточного и БПХ слоев равно $\frac{f^2 m}{\log s}$. Следовательно, добавление слоя быстрого преобразования Хафа вносит небольшой вклад в сложность сети, который тем меньше, чем сложнее сверточная часть нейронной сети. Однако преобразование Хафа по всему диапазону углов увеличивает площадь карт признаков примерно в 8 раз, что, безусловно, скажется на времени выполнения сверточных слоев по ним. Для решения этой проблемы запланировано исследование, связанное с преобразованием признаков, полученных с помощью слоев БПХ из (s, t) координат в (ρ, φ) координаты, которое позволило бы задать шаг сдвига и угла и подобрать оптимальный размер карт признаков. Это исследование в диссертацию не входит.

3.6 Перенос знаний

Внимательное изучение свойств карт признаков после слоя прямого БПХ позволяет заметить, что для человека они визуаль-но неразличимы при решении схожих задач на данных различной природы [77]. Такая гипотеза хорошо согласуется с аналогичным вопросом при решении задачи инженерным алгоритмом: после первого БПХ, если мы, например, ищем несобственную точку схода, на карте признаков находятся пики интенсивности в местах, соот-

ветствующих различным прямым на исходном изображении, а сам способ получения этих прямых уже не играет роли, т.е. получается, что, извлекая из разных данных прямые разным способом, на полученных признаках точку схода можно далее искать общим алгоритмом.

Возвращаясь к нейронной сети со слоями БПХ, можно предположить, что большая часть информации, которая специфична для конкретных данных, выделяется именно первым блоком сверток, а остальная часть сети уже работает на признаках более высокого уровня и, на самом деле, о природе данных ничего и не знает. Получается, что если решать схожие задачи (например, поиск несобственной точки схода) на данных разной природы, то можно для разных данных построить разные наборы сверток для первого блока сети, а дальнейшие слои обучать уже на смешанных данных, тем самым увеличив обучающую выборку и ее общность путем переноса знаний из одного домена данных в другой.

Для проверки этой гипотезы проведем серию экспериментов по поиску несобственной точки схода, а в качестве двух доменов данных возьмем изображения документов и изображения дорог с видеорегистраторов. Так как во втором наборе данных точка схода собственная, то они будут переделаны таким образом, чтоб она стала внешней. Схема переделывания данных изображена на рисунке 3.7. Сначала находится прямоугольная зона на изображении, для которой имеющаяся точка схода является внешней, и проективно трансформируется к исходному размеру.

Для всех изображений была создана разметка, состоящая из нескольких прямых, образующих искомый пучок. Примеры комплектов таких изображений для документов (сверху) и дорог (снизу) показаны на рисунке 3.8. Все изображения имеют размер 300×300 , а точка схода удалена от края не менее, чем на 100 пикселей. В наборе изображений с документами в общей сложности 12800 изображений, а с дорогами 3800 изображений. 90% последних в данной задаче будем рассматривать целевым набором данных, на

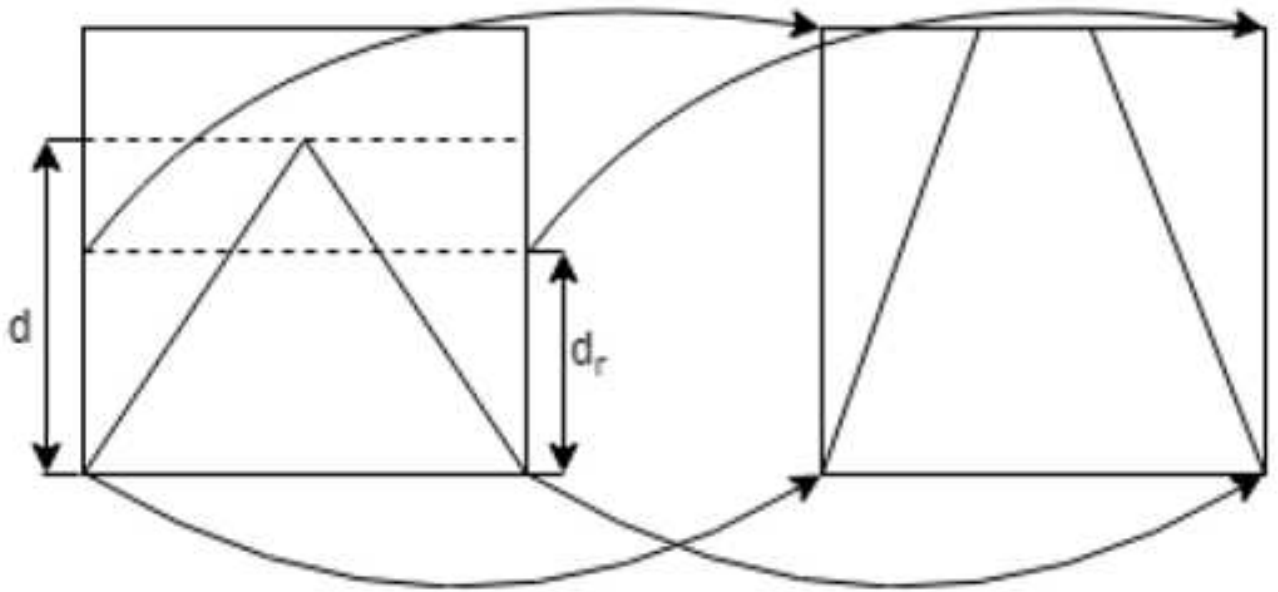


Рисунок 3.7 — Способ перевода собственной точки схода в несобственную для изображений дорог.

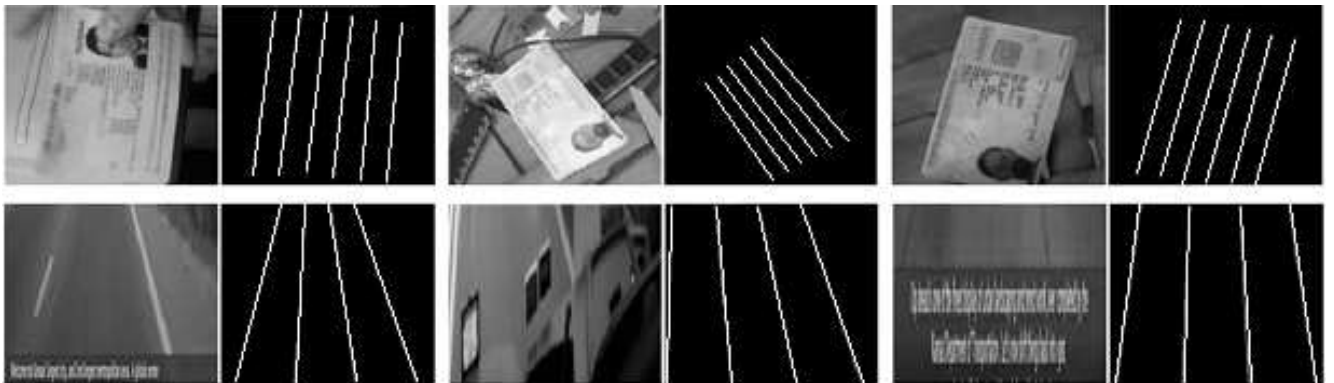


Рисунок 3.8 — Примеры изображений с соответствующей разметкой для документов (сверху) и для дорог (снизу).

котором необходимо решить задачу поиска вертикальной точки схода, а 10% будем использовать для обучения.

Для решения этой задачи была создана нейронная сеть на базе архитектуры Хафнет, подробно описанной в параграфе 3.2, состоящая из трех блоков сверточных слоев, разделенных слоями быстрого преобразования Хафа [73]. В этой сети в общей сложности 3.1×10^4 обучаемых параметров, но только 7.5×10^3 из них находятся в первом блоке сверточных слоев.

Ошибкой в данном случае будем считать отклонение найденной точки схода от ее идеальной позиции. Учитывая размер

изображения и удаленность точки можно говорить, что ошибка в 15 пикселей является приемлемой.

В итоге было проведено три эксперимента. В первом эксперименте нейронная сеть обучалась на обучающей части изображений дорог. Во втором эксперименте сеть училась на изображениях документов, а в третьем эксперименте для проверки гипотезы о возможности переноса знаний параметры для второго и третьего блоков сверточных слоев были взяты из сети, обученной на документах, а параметры первого блока были дообучены на обучающей части изображений дорог. Результаты работы этих трех сетей на обучающей и на тестовой частях набора данных с дорогами сведены в таблице 7.

Таблица 4 — Сравнение ошибки в трех экспериментах

Эксперимент	Ошибка на обучающих данных	Ошибка на тестовых данных
Первый	14.8	17.2
Второй	24.2	23.8
Третий	15.3	12.6

Наивное обучение всей сети на 10% данных, т.е. на 380 изображениях (первый эксперимент) ожидаемо не дает хорошего качества, так как этого очень мало для обучения. Кроме этого, ошибка на обучающей части заметно меньше, чем на тестовой, что говорит о том, что сеть переобучилась из-за недостатка данных. Сеть, обученная целиком на документах ожидаемо имеет высокую ошибку на обеих частях набора данных с дорогами (второй эксперимент). А вот дообучение лишь части сети на небольшом количестве релевантных изображений (третий эксперимент) дает минимальную ошибку и наглядно демонстрирует факт переноса знаний между доменами данных изображений документов и дорог.

3.7 Результаты главы 3

В данной главе разработаны архитектуры нейронных сетей, использующих предложенные ранее слои прямого и транспонированного быстрого преобразования Хафа. Для каждой из архитектур описаны способы ее применения и случаи, когда именно ее применение обосновано, исходя из конфигурации слоев и комбинации прямого и транспонированного быстрого преобразования Хафа.

1. Предложена архитектура Хафнет, содержащая в себе ветви с последовательными прямыми быстрыми преобразованиями Хафа, разделенными свертончными слоями. Такая архитектура сети позволяет находить внешние точки схода, в том числе и бесконечно удаленные благодаря применению двух последовательных быстрых преобразований Хафа.
2. Для архитектуры Хафнет разработан метод отображения результата нейронной сети конечного размера на бесконечную плоскость, включающую исходное изображение, с помощью кусочнопроективного преобразования. Главным преимуществом такого отображения является способность описывать бесконечно удаленные точки схода. Благодаря этому их можно обрабатывать обычными методами.
3. Предложена архитектура Хафэнкодер с описанием границ применимости и принципа работы. Такая сеть позволяет получить результат в исходных координатах, благодаря включению в себя слоя транспонированного быстрого преобразования Хафа. Такая сеть хорошо подходит для многих задач семантической сегментации и поиска внутренних точек схода.
4. Показано, что признаки в пространстве (s, t) имеют общий вид для входных изображений различной природы, что позволяет расширить обучающие данные для конкретной задачи данными из других областей. Также это позволяет использовать перенос знаний, накопленных в последних сло-

ях (в отличие от стандартного переноса, где, как правило, участвуют ранние слои), с одной задачи на другую. Проведен подтверждающий это эксперимент с использованием двух существенно различающихся наборов данных (изображения документов и изображения с видеорегистратора автомобиля).

Глава 4. Практические применения нейронных сетей разработанных архитектур

В этом разделе будут рассмотрены способы применения нейронных сетей разработанных архитектур с использованием слоев прямого и транспонированного быстрого преобразования Хафа, описанных в главе 3. Такие сети способны учитывать интегральные признаки вдоль линейчатых элементов на изображении и, следовательно, особенно хорошо подходят для решения ряда задач.

4.1 Поиск внутренней точки схода полносвязной сетью с БПХ

В качестве первого эксперимента для поиска точки схода рассмотрим задачу поиска точки схода на изображениях, полученных с камеры видеорегистратора транспортного средства [78]. Такие изображения, как правило, содержат единственную точку схода, находящуюся внутри изображения, как показано на рисунке 4.2.

В классических алгоритмах обработки изображений ключевым моментом в поиске точки схода является использование преобразования Хафа два раза. Как известно, при таком преобразовании прямые переходят в точки, а прямые, которые пересекаются в одной точке на исходном изображении переходят в точки, лежащие на одной прямой. Максимум на результате второго преобразования будет соответствовать точке схода.

Так как в референсной работе [39] эта задача решалась путем покрытия исходного изображения решеткой и определением ячейки, в которой находится точка схода, то для корректного сравнения мы делаем также. Сперва мы попробовали использовать предложенную архитектуру AlexNet, но впоследствии поняли, что заявленное качество может быть достигнуто и более простой архитектурой нейронной сети. Архитектура нейронной сети состояла из 4 сверточных

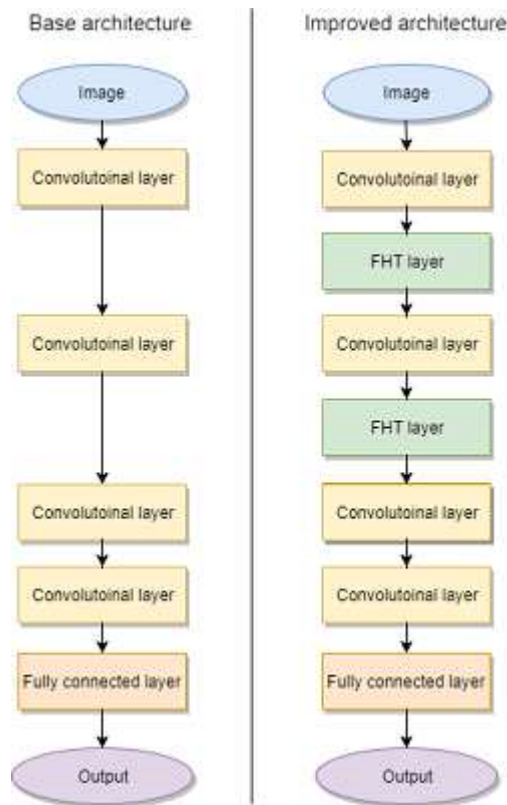


Рисунок 4.1 — Стандартная и модифицированная архитектуры.

слоев и одного полносвязного. Главное отличие нашей базовой архитектуры от AlexNet состоит в меньшем количестве фильтров в сверточных слоях и всего лишь одном полносвязном слое.

После обучения базовой сети была обучена сеть с новой архитектурой. Главным нововведением было присутствие двух слоев БПХ, которые мы добавили между сверточными слоями. Схемы базовой и улучшенной архитектур представлены на схеме 4.1. Стоит отметить, что для обучения сетей мы использовали сильно измененную версию CUDA-Convnet [79].

В результате данной работы был реализован улучшенный алгоритм обучения сверточной нейронной сети через слои быстрого преобразования Хафа. Хотя этот вид слоя может быть определен как полносвязный и теоретически здесь не нужно ничего изобретать, матрица такого слоя будет состоять из n^2 элементов, где $n = w * h * c$ — длина входного вектора. Например, для изображения размером $100 \times 100 \times 20$ (относительно небольшого для сверточной нейронной сети) матрица весов будет иметь $4 * 10^{10}$

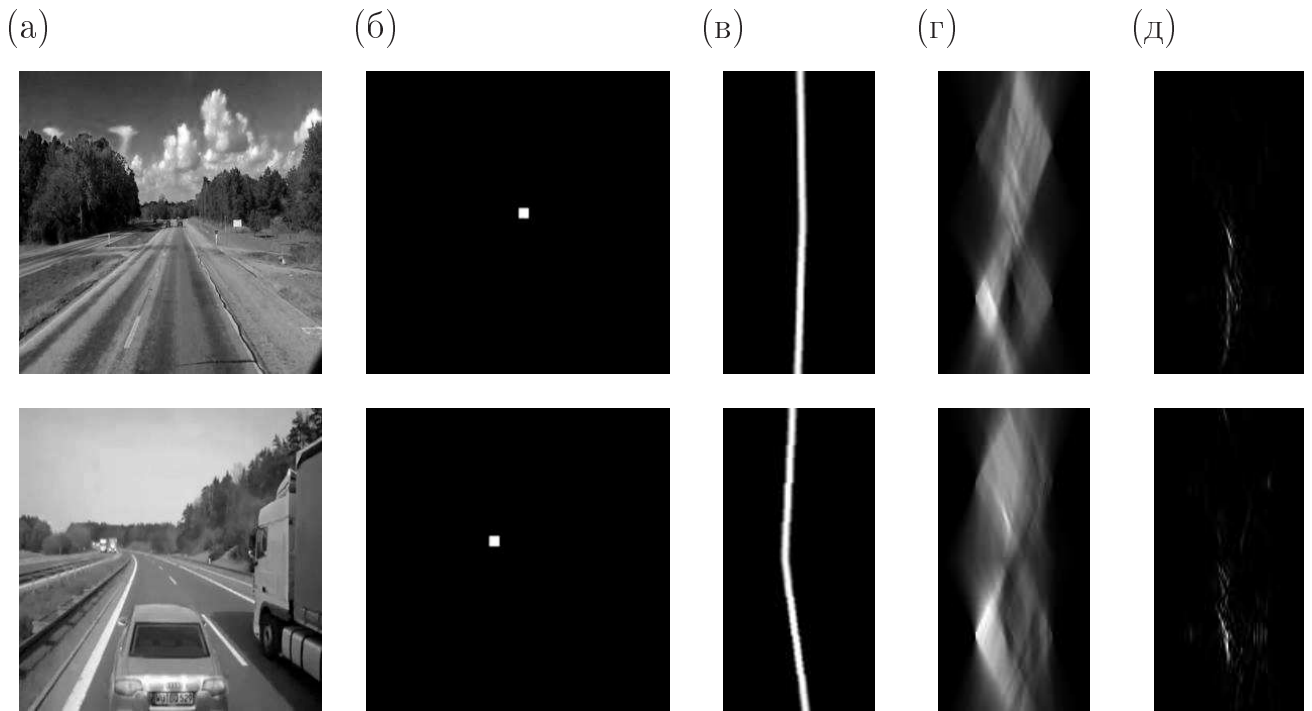


Рисунок 4.2 — Примеры из нейронной сети: а) Входное изображение; б) Идеальное изображение; в) БПХ идеального изображения; г) БПХ исходного изображения; д) БПХ исходного изображения после применения обученных сверточных слоев.

элементов. Реализованный подход позволяет посчитать прямой и обратный проходы через слой БПХ без использования такого вида матриц.

На рисунке 4.2 показаны примеры работы предложенного алгоритма и наглядно представлено назначение слоя БПХ. Рис. 4.2 а) показывает исходные изображения; на б) представлено идеальное изображение – белая точка на черном фоне в месте предполагаемой точки схода, в) – идеальная линия, на которой располагаются все точки в Хаф-пространстве, соответствующие линиям, проходящим через точку схода на идеальном изображении. г) – БПХ исходного изображения, д) – БПХ исходного изображения после обученных сверточных слоев. Видно, что с использованием предложенного метода можно обучить сверточные слои выделять подходящие прямые и границы и после этого решать задачу.

Еще раз отметим, что у слоя БПХ нет обучаемых коэффициентов, поэтому он не усложняет архитектуру сети в плане количества

весов. Все операции на этом слое предопределены и не меняются в процессе обучения.

4.2 Задачи, решаемые в процессе обработки изображений документов

Цифровая обработка изображений документов с целью извлечения из них информации – важная составляющая многих процессов. Например, документ часто нужно найти на изображении и исправить проективное искажение для того, чтоб эффективно сериализовать его или для того, чтоб распознать текстовые поля. Поиск определенных ригидных объектов на уже найденном документе – часто возникающая задача. В этой секции рассмотрим различные задачи, возникающие при цифровой обработке документов, которые можно решать с помощью нейронных сетей, содержащих в себе слои прямого и транспонированного быстрого преобразования Хафа.

4.2.1 Поиск внешних точек схода для документов

Рассмотрим задачу определения ориентации документа на изображении. Такая задача естественным образом возникает, например, при оцифровке данных с документов, так как определенные позиция и ориентация позволяют найти нужный регион изображения, привести документ к прямоугольной форме и сохранить или распознать его, так как в таком виде это делать проще и надежней. Для того, чтоб найти ориентацию объекта можно либо найти его края и сам четырехугольник, либо же найти точки схода. Первое не всегда возможно, так как края могут быть заслонены какими-либо объектами на изображении или же вообще

выходить за его пределы. Точки схода в таком случае можно все равно найти, так как они образованы строками текста, началами выравненных текстовых полей, даже элементами букв. Таким образом, изображение документа содержит две точки схода (вертикальная и горизонтальная). Геометрическое место таких точек в обычном случае лежит далеко за пределами картинки вплоть до бесконечности. Чем меньше прямоугольник документа подвержен проективным искажениям, тем дальше эти точки находятся.

Такая задача может быть решена с помощью нейронной сети с архитектурой Хафнет, описанной в разделе 3.2. Эта сеть на выходе имеет две двумерные карты признаков. Используя уравнения 3.3 и 3.4 можно позиции максимумов на соответствующих выходах перевести в позиции точек схода в координатах исходного изображения.

Для обучения нейронной сети и ее тестирования было использовано два разных набора данных, находящихся в открытом доступе. Первый из них MIDV-500, он состоит из изображений документов [80]. Эти изображения были получены с помощью камер различных мобильных устройств и, следовательно, содержат специфические искажения, такие как проективное искажение. Несколько примеров таких изображений показано на Рис. 4.3. Сами документы обычно сделаны из пластика или плотного картона и поэтому сохраняют плоскую прямоугольную форму. Таким образом, этот набор данных идеально подходит для поставленной задачи. Первые 30 из 50 типов документов были использованы для обучения (7217 изображений), а остальные 20 были использованы для тестирования (4748 изображений). При таком разбиении не существует такого типа документа, изображения которого бы одновременно попали и в обучающий, и в тестовый наборы данных. Кроме этого, все изображения, на которых больше одного угла документа выходит из зоны видимости были убраны. Дополнительно, изображения были отмасштабированы так, чтоб их ширина равнялась 400 пикселей.

Так как на первом наборе данных нет опубликованных результатов в рассматриваемой задаче и, следовательно, невозможно количественно оценить предложенный подход, был использован



Рисунок 4.3 — Примеры из набора данных MIDV-500.

второй набор данных. Это набор ICDAR 2011, который использовался в соревновании по выпрямлению документов [81]. Эти изображения были отмасштабированы так, чтобы их ширина составляла 500 пикселей. Особенностью этого набора является то, что кроме проективных искажений, в нем еще представлены и различные изгибы, не описываемые мелко-линейными преобразованиями. И хотя предлагаемый метод идеологически не должен уметь такое исправлять, его устойчивость к наличию подобных искажений также имеет высокую ценность и гарантирует его применимость в реальных приложениях, где данные всегда страдают сразу множеством несовершенств. Примеры таких изображений показаны на Рис. 4.4. Как можно видеть, изображения из этих двух наборов данных существенно разные, но, даже будучи обученной на первом наборе данных, нейронная сеть описываемой архитектуры способна показать более высокое качество на втором наборе, что доказывает ее высокую обобщающую способность.

Проведенный эксперимент состоял из двух частей. В первой части обученная нейронная сеть была применена к тестовой части набора данных MIDV-500 и оценено качество определения несобственных точек схода. Для этого, с помощью найденных точек схода, изображение документа было приведено к прямоугольному и затем оценена степень его прямоугольности. Чем лучше были найдены точки схода, тем ближе к прямоугольнику должен получиться документ. Для численной оценки было подсчитано два расстояния: d_1 и d_2 согласно следующим формулам (N – количество изображений):

Таблица 5 — Результаты на наборе данных MIDV-500.

	Train	Train corrected	Test	Test corrected
$d_1(^{\circ})$	2.56	1.63	2.70	1.65
$d_2(^{\circ})$	5.76	0.93	4.77	0.91

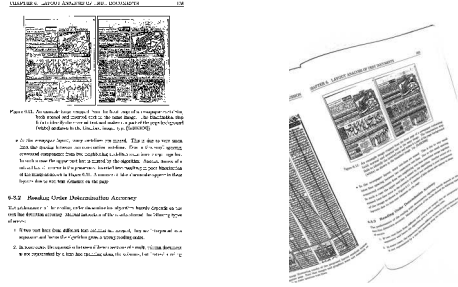


Рисунок 4.4 — Примеры изображений из набора данных ICDAR 2011.

$$d_1 = \frac{1}{4N} \sum_{j=0}^{N-1} \sum_{i=0}^3 |90^{\circ} - \alpha_i^j| \quad (4.1)$$

$$d_2 = \frac{1}{2N} \sum_{j=0}^{N-1} |\alpha_v^j| + |\alpha_h^j| \quad (4.2)$$

Расстояние d_1 показывает среднее отклонение углов исправленного документа от 90° независимо от его ориентации, как видно из формулы 4.1. Расстояние d_2 , наоборот, показывает насколько хорошо была исправлена ориентация документа независимо от степени его прямоугольности, как можно видеть из формулы 4.2. Чем меньше эти значения, тем точнее была проведена корректировка формы и ориентации. В таблице 5 представлены результаты и для исходных, и для исправленных изображений документов, чтоб можно было оценить эффект от исправления. Аналогичные результаты для обучающего набора данных здесь приведены для демонстрации того, что переобучения не произошло: качество работы сети примерно одинаковое.

Вторая часть эксперимента была проведена для демонстрации высокого качества в сравнении с опубликованными результатами и для доказательства сильной обобщающей способности предлагаемого решения. Для этого изображения из набора данных (100

Таблица 6 — Результаты на наборе данных ICDAR 2011.

Данные	Исходные	[83]	[84]	Предлагаемый метод
Распознано (%)	31.3	49.6	50.1	59.7

бинарных изображений страниц с текстом в различных вариантах заполнения, таких как одна или две колонки, наличие рисунков или таблиц) по соревнованию в выпрямлении изображений ICDAR 2011 были в различных видах поданы на вход нейронной сети и исправлены. Затем, для оценки качества исправления, на них был запущен Tesseract OCR версии 3.02 [82]. Сравнение результатов проводилось с опубликованными в работах [83, 84] значениями. Из полученных результатов, которые показаны в таблице 6, можно видеть, что даже будучи обученной на совсем других данных, нейронная сеть справилась с задачей и показала лучшее качество. Такое возможно благодаря особенностям архитектуры: хоть сами изображения и разные, но ключевые элементы, которые и используются нейронной сетью для поиска ответа схожи (вертикальные штрихи букв, строки, начала строк). Важно заметить, что такие признаки не инварианты относительно масштабирования, а значит необходимо знать их примерный размер для успешного применения метода.

На Рис. 4.5 показан процесс выпрямления изображения с помощью нейронной сети предложенной архитектуры. Первый столбец содержит входные изображения, второй – вход первого БПХ слоя (тут изображена только одна карта признаков из ветви для горизонтальных точек схода для удобства визуализации). Третий столбец аналогично показывает вход второго БПХ слоя и последний, четвертый, показывает результирующее исправленное с помощью найденных точек схода изображение. Изображения с ветви для горизонтальных точек схода были выбраны для визуализации? так как для этих точек признаков больше (горизонтальные строки текста) и, следовательно, иллюстрации нагляднее.

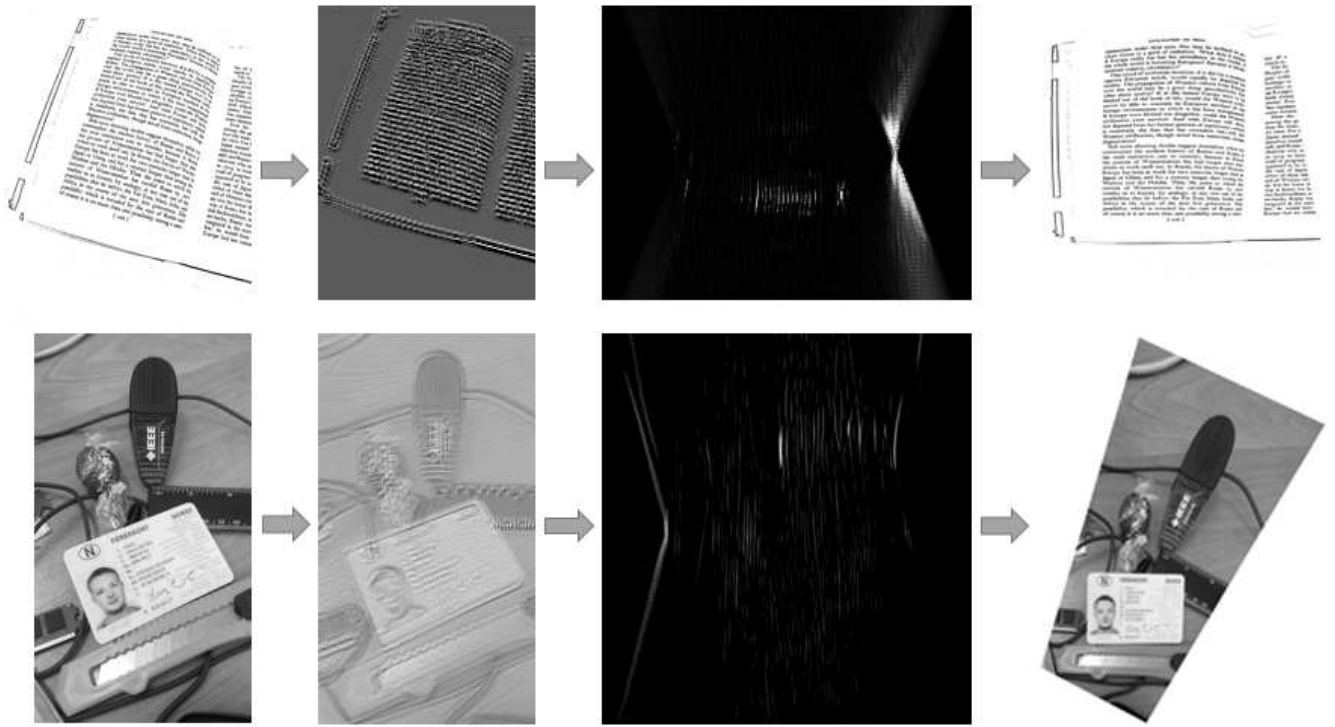


Рисунок 4.5 — Процесс выправления документа. Сверху – ICDAR 2011, снизу – MIDV-500.

4.2.2 Семантическая сегментация изображений документов

В параграфе 3.4 был проведен эксперимент, демонстрирующий возможности архитектуры нейронной сети со слоями прямого и транспонированного БПХ. Конечно, там рассматривался сильно упрощенный случай, который, тем не менее, дает основания полагать, что подобной архитектуре будет заметно проще справиться с некоторыми видами задачи семантической сегментации.

Рассмотрим задачу семантической сегментации изображений документов. Как и в рассмотренном ранее примере будем учить сеть таким образом, чтоб она могла каждый пиксель классифицировать как принадлежащий документу или же принадлежащий фону.

Для эксперимента был использован набор данных MIDV-500 [80]. Это открытый набор данных, содержащий раскадровки коротких видео. На этих видео снимались различные документы 50 типов в разных условиях в том числе со сложными фонами. Дополнительным бонусом данного набора данных является то, что видео

с документами разных типов разделены друг относительно друга и возможно построить тестовый и обучающий наборы так, чтобы они не пересекались по использованным типам документа. Первые 30 типов документов были отнесены в обучающую выборку, а вторые 20 – в тестовую. Все изображения были переведены к серому путем усреднения интенсивностей по каналам и отмасштабированы к размеру 256×256 . Общее количество изображений в обучающей выборке – 7217, а в тестовой – 4748.

Дополнительно был использован еще один схожий набор данных MIDV-2019 [85]. Это схожий набор данных, но документы на изображениях подвержены более сильным проективным искажениям. Для того, чтоб представить искажения такой силы в обучающем наборе данных, туда были добавлены изображения одного типа документов (120 изображений).

Для тестирования было обучено три нейронных сети: Хафэнкодер (обозначен “HE”), полносверточная сеть с конфигурацией сверток как у Хафэнкодера, но без слоев прямого и транспонированного БПХ (обозначен “HE*”) и стандартная UNet сеть. Все три нейронных сети обучались в одинаковых условиях на одинаковых данных: изображения 30 типов документов из MIDV-500 и 1 типа документов из MIDV-2019. При обучении для всех трех использовались одни и те же раздутия на лету [86] и каждая сеть училась по 200. Дополнительно, данный эксперимент был проведен по пять раз для каждой сети и взято среднее качество для большей достоверности.

Оценка результата проводилась с помощью оценки среднего отношения пересечения найденных и идеальных зон документа к их объединению, MIoU (mean Intersection over Union), вычисляемого по формуле 4.3.

$$MIoU = \frac{1}{N} \sum_{i=0}^{N-1} \frac{A_i \cap G_i}{A_i \cup G_i}, \quad (4.3)$$

В формуле 4.3 A_i – найденная зона документа, G_i – истинная зона документа, N – общее количество изображений.

Таблица 7 — Сравнительные результаты, MIOU

Сеть	Количество параметров	Качество MIDV-500	Качество MIDV-2019
UNet	1.96×10^6	0.94	0.86
HE*	1.48×10^4	0.91	0.74
HE	1.48×10^4	0.96	0.89

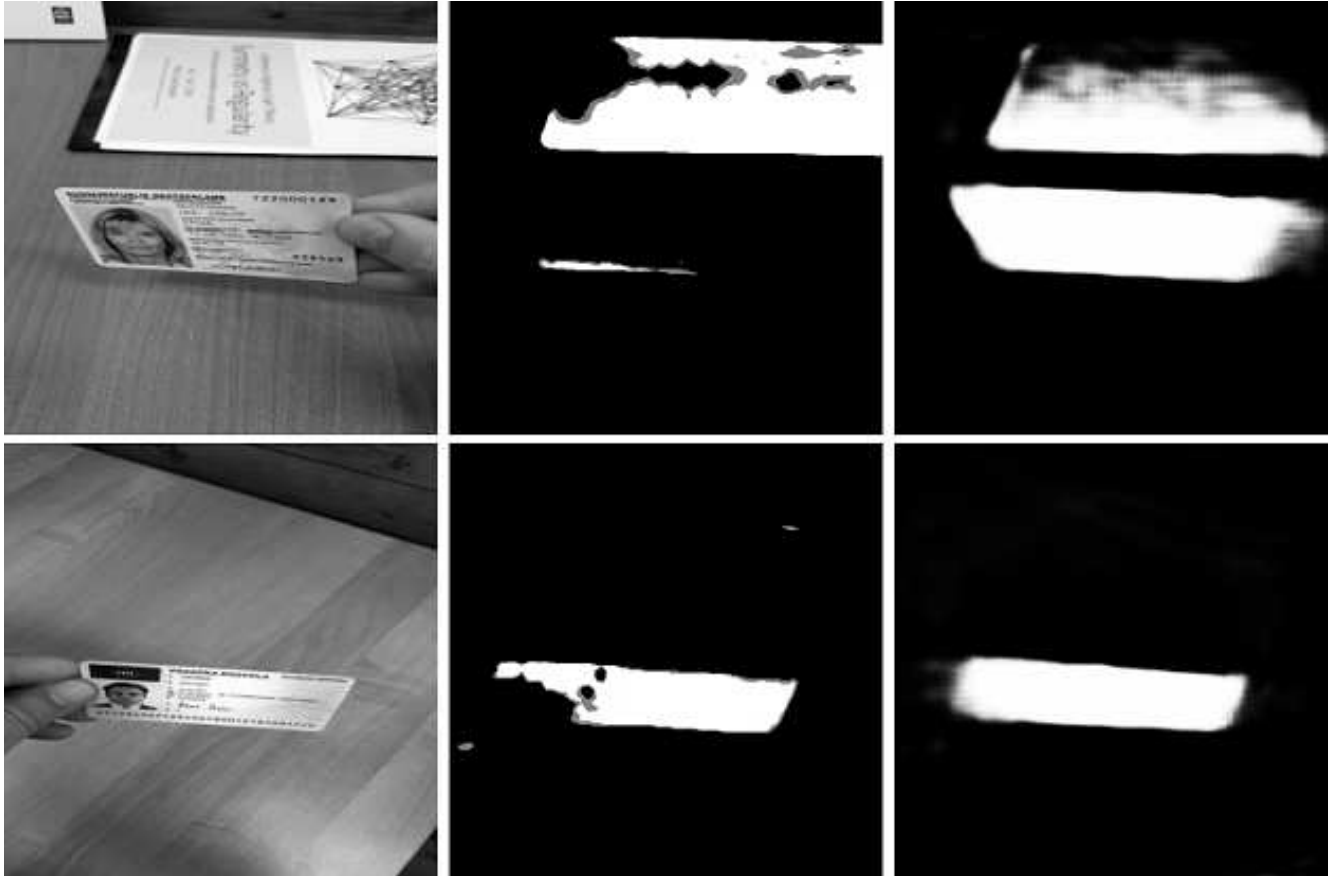


Рисунок 4.6 — Сравнение работы Хафэнкодера и UNet в сложных случаях. Входное изображение (слева), результат UNet (в центре), результат Хафэнкодера (справа).

Сравнительные характеристики с точки зрения количества обучаемых параметров и полученного качества приведены в таблице 7. Из результатов можно видеть, что при в 132 раза меньше количестве обучаемых параметров Хафэнкодер все равно показывает большее качество, чем стандартная полносверточная архитектура. Дополнительно можно видеть, что полносверточная сеть, равная по количеству обучаемых параметров Хафэнкодеру показывает, наоборот, худший результат.



Рисунок 4.7 — Примеры изображений документов с машиночитемой зоной.

На изображении 4.6 приведен пример работы Хафэнкодера и Unet на некоторых сложных случаях. На примере сверху на фоне находится объект, фактически являющийся вторым документом, но которого не было в разметке. Интересно, что UNet здесь показала неадекватный результат, в то время как Хафэнкодер с некоторой уверенностью подсветил обе сущности, что в целом верно. В примере снизу показан документ, целостность которого нарушают фотография и палец в комбинации. Видно, что UNet не справился с закраской именно этого региона, в то время как Хафэнкодер выдержал четырехугольность зоны.

4.2.3 Использование Хафэнкодера для поиска МЧЗ

Другой задаче в распознавании документов является поиск машиночитаемой зоны (МЧЗ, MRZ, Machine Readable Zone). Эта зона напечатана специальным шрифтом и содержит в себе основную информацию документа для того, чтоб можно было в автоматическом режиме быстро и надежно ее прочесть. Примеры документов с такой зоной представлены на рисунке 4.7.

Машиночитаемая зона состоит из двух или трех сравнительно длинных строк текста. Учитывая, что на изображениях документов часто бывают блики или заслоняющие объекты, то для корректного поиска такой зоны необходимо видеть ее всю, т.е. иметь рецепивное поле, сравнимое с размером изображения. Напротив, Хафэнкодер дает нам механизм учета интегральных признаков с использовани-

ем обычных сверточных слоев. Таким образом, этот объект хорошо подходит для того, чтоб его искать с помощью одной из разработанных архитектур: HoughEncoder [87].

Для проведения эксперимента и корректного сравнения с имеющимися результатами был использован ранее опубликованный набор синтетических данных [88]. В этом наборе данных содержится 456000 изображений документов, взятых из интернета (Google Images), которые помечены, как свободные к использованию и распространению. Случайно сгенерированный текст (но удовлетворяющий требованиям стандарта) для машиночитаемой зоны был помещен в нижнюю часть документа с помощью методов обработки изображений, а само изображение документа было помещено на случайный фон. К финальным изображениям для имитации натуральности были применены различные преобразования для моделирования искажений, соответствующих тем, что появляются на натуральных фотографиях документа: проективное искажение, аддитивный шум, размытие. Примеры изображений из этого набора данных представлены на Рисунке 4.7.

Для сравнения качества работы на этом наборе данных нейронная сеть должна вернуть карту семантической сегментации (бинарную маску), где каждый пиксель был бы отнесен либо к машиночитаемой зоне, либо к фону (здесь фоном считается все, что не попадает в первый класс, т.е. другие пиксели документа тоже фон). После получения такой карты вызывался алгоритм сбора компонент связности и распознавания из оригинальной работы [88], результатом которого являлось качество распознавания строк. Эта часть алгоритма была неизменной для всех трех вариантов поиска карты семантической сегментации: предлагаемый Хафэнкодер, основанный на бинаризации метод [89], основанный на особых точках YAPE метод [90].

Для обучения Хафэнкодера было случайно взято 10% изображений из использованного набора данных и вручную создана для них разметка позиции строк машиночитаемой зоны. Все изображения были приведены к размеру 180×180 , а значения по



Рисунок 4.8 — Извлечение изображения машичитаемой зоны: исходное изображение, карта семантической сегментации, наложенная на изображение карта, зоны строк, машиночитаемая зона, извлеченное изображение машиночитаемой зоны.

каналам были усреднены. Результирующий размер изображения был подобран эмпирически исходя из двух соображений. Первое: если на изображении человек легко находит машиночитаемую зону, то должна справиться и сеть, и второе: чем меньше размер изображения, тем меньше на нем информации и, следовательно, тем проще решить задачу в случае, если искомые данные не потеряны. Для обучения сети использовалась mIOU функция потерь (mean Intersection Over Union). Для улучшения обучения к данным применялись раздутия на лету, которые представляли собой базовые преобразования изображения: проективные искажения, поворот, сдвиг. Благодаря таким искажениям позиция искомой зоны на исходном изображении была менее предсказуемой. Нейронная сеть обучалась 660 эпох с размером минибатча 64.

Таким образом, алгоритм состоит из следующих шагов: поиск карты семантической сегментации, анализ компонент связности в ней, наложение на исходное изображение и извлечение изображения машиночитаемой зоны. Схематично весь процесс изображен на Рисунке 4.8. Полученное изображение машиночитаемой зоны подавалось на вход алгоритму сегментации и распознавания и измерялось результирующее качество. Результаты сравнения методов поиска машиночитаемой зоны приведены в таблице 8.

Таблица 8 — Сравнительные результаты распознавания машиночитаемой зоны при разных методах поиска.

Метод поиска	Метод сегментации	Качество, %
Базовый [88]	Базовый [88]	56.0
С YARE [90]	Базовый [88]	94.2
С бинаризацией [89]	Базовый [88]	97.8
Предлагаемый (Хафэнкодер)	Базовый [88]	98.1

4.3 Поиск внутренней точки схода на изображениях дороги

В этой секции опишем решение задачи поиска внутренней точки схода на изображениях дорог с помощью нейронной сети с архитектурой, описанной в секции 3.3.

Для проведения эксперимента использовались данные, предоставленные авторами работы [39]. Эти данные содержат в себе изображения, полученные с видеорегистраторов транспортных средств в различных условиях (городские дороги, междугородние трассы, проселочные дороги). Исходные видео были взяты авторами с видеохостинга YouTube и разбиты на последовательности кадров. Размеры изображений составляют 300×300 пикселей. Всего набор данных содержит 9972 изображений, которые были случайным образом разделены на обучающую (8974 изображений) и тестовую



Рисунок 4.9 — Примеры изображений.

(998 изображений) выборки. Примеры таких изображений показаны на Рис. 4.9. Для оценки качества предложенного решения была использована методика оценки, описанная авторами набора данных: изображение разбивалось на 10×10 , 20×20 и 30×30 ячеек и считалось, что ответ правильный, если он попадает в правильную ячейку.

В результате был получен улучшенный алгоритм обучения нейронной сети со слоями прямого и транспонированного преобразования Хафа. Благодаря использованию быстрого алгоритма и для транспонированных версий оператора, вычислительная сложность была понижена с (n^3) до $O(n^2 \log(n))$.

На рисунке 4.10 показаны примеры работы предложенного алгоритма и визуализирован смысл БПХ слоя. Так как промежуточные карты признаков многоканальные, для иллюстраций были выбраны отдельные каналы, который казались информативнее. На (a) показано исходное изображение, на (b) карта признаков после первого блока сверточных слоев, на (c) результат слоя БПХ, на (d) выход второго блока сверточных слоев, на (e) выход транспонированного БПХ слоя и, наконец, на (f) показан результат работы сети. Благодаря такой архитектуре, нейронная сеть способна выбирать релевантные прямолинейные участки на входном изображении и объединять их в точки схода.

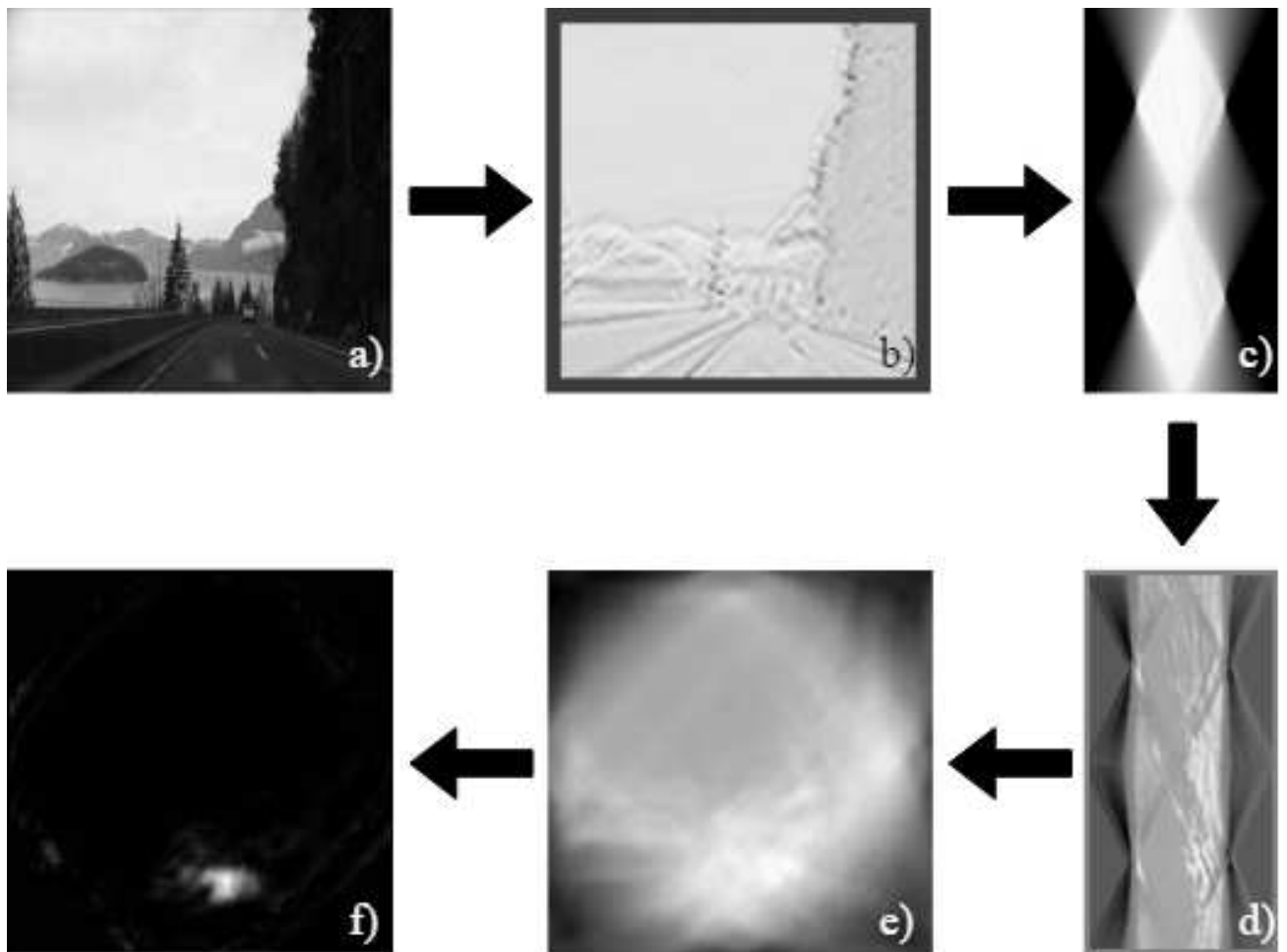


Рисунок 4.10 — Примеры карт признаков с разных слоев нейронной сети: а) вход сети; б) выход первого блока сверточных слоев; с) выход прямого БПХ; д) вход транспонированного БПХ; е) выход транспонированного БПХ; ф) выход сети.

В таблице 9 показаны результаты первого эксперимента для базовой (т.е. взятой из опубликованной работы) и предложенной архитектур нейронных сетей для различных размеров решетки и количества рассматриваемых альтернатив ответа. Предложенная сеть показывает большее качество и, что очень важно, ее качество практически не зависит от размера решетки в отличие от сети с полносвязным слоем. Это достигается благодаря ее полносверточной природе и сильно меньшей склонности к переобучению.

Для демонстрации превосходства предложенного метода и исследования его устойчивости к различным искажениям изоб-

Таблица 9 — Сравнение ошибок представленного метода с базовым

Размер решетки	Базовый топ 1, %	Базовый топ 5, %	Предложенный топ 1, %	Предложенный топ 5, %
10 × 10	31.7	5.2	1.5	0.7
20 × 20	44.5	16.1	5.4	2.3
30 × 30	52.9	25.6	6.2	3.1



Рисунок 4.11 — Примеры добавленных в изображения искажений, призванных сделать невозможным локальный поиск точки схода.

ражения был проведен дополнительный эксперимент. Для этого без переучивания нейронной сети был модифицирован тестовый набор данных следующим образом: прямоугольные зоны разного размера вокруг искомой точки схода были размыты гауссовым сглаживанием так, чтобы их невозможно было различить исходя из локальной окрестности изображения. Примеры таких модифицированных изображений показаны на рисунке 4.11.

Результаты этого эксперимента частично представлены на рисунке 4.12. На графиках показана зависимость ошибки от размера размытой области вокруг точки схода для различных размеров решетки. Легко можно видеть, что предложенная нейронная сеть, вообще говоря, не требует наличия области изображения, на котором находится непосредственно сама точка схода. Это демонстрирует одно из главных свойств этой сети: способность в каждой локальной точке принимать решение, исходя из глобальных признаков, без использования дополнительных обучаемых параметров.

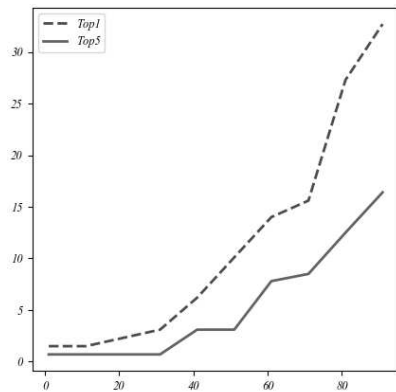
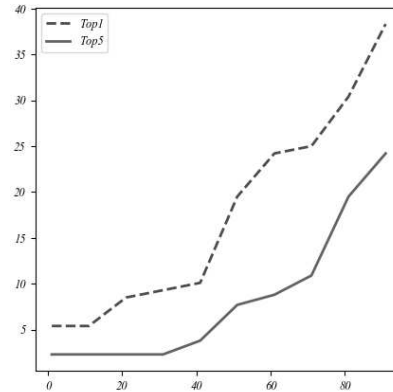
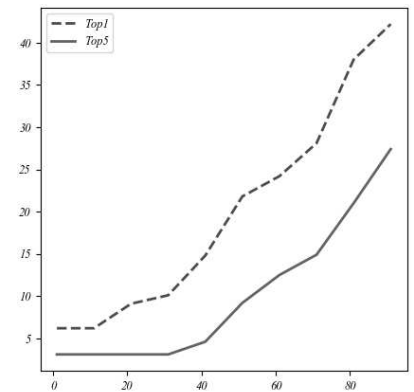
(а) 10×10 (б) 20×20 (в) 30×30 

Рисунок 4.12 — Деградация качества при порче изображения для решеток различного размера.

Более того, рассматриваемый случай, когда точка схода может оказаться закрыта, часто встречается в реальных ситуациях: она может быть загорожена зданием, другой машиной или просто физически отсутствовать из-за рельефа местности. В таких случаях стандартные методы оказываются полностью неприменимыми, когда предложенный все еще может справиться с задачей. Таким образом, использование слоев прямого и транспонированного преобразования Хаф в нейронной сети не только улучшает ее качество и уменьшает количество обучаемых параметров, но еще и существенно расширяет область применения путем добавления принципиально новых качеств.

4.4 Практическое применение

4.4.1 Общее описание системы обучения

Для проведения всех экспериментов за базовую систему обучения была взята модифицированная версия CudaConvnet [79]. Эта система хороша тем, что при практически полном наборе

функциональности в плане обучения сетей прямого распространения ошибки имеет сравнительно небольшой и простой исходный код. Это делает ее легкой для понимания и модифицирования, а эксперименты на базовых открытых наборах показывают, что state-of-the-art качество достигается. Также эта система была использована и показала свою эффективность во многих смежных работах научного коллектива [91, 92, 93]. В самой системе нейронная сеть реализованна как массив объектов типа “слой”, каждый из которых обладает своими параметрами и переменными, включая входные данные для сети и функции потерь. Фактически, это является двусвязным списком слоев, благодаря чему удобно обходить сеть, даже если она состоит из многих ветвей.

Данная система обучения использовалась и продолжает использоваться для обучения нейронных сетей научной группой лаборатории 9.2 ФИЦ ИУ РАН, а также компанией Smart Engines Service в коммерческих целях. Имеется свидетельство о регистрации программы для ЭВМ [94]. Кроме этого, автором диссертации и коллективом выпущено множество публикаций, для достижения результатов в которых была использована разработанная система обучения нейронных сетей. Достигнутое state-of-the-art качество во многих задачах говорит о высоком уровне работы системы и ее соответствии заявленному качеству.

В эту систему был добавлен слой быстрого преобразования Хафа со следующими параметрами: диапазон углов, использование выравнивания, направление. Все эти параметры подробно были описаны в главе 2.

4.4.2 Система SmartID Reader

Разработанные нейронные сети в качестве практического внедрения встраивались в систему распознавания SmartID Reader, разработанную организацией Smart Engines Service. Это система

распознавания документов, хотя только распознаванием ее функционал не ограничивается. Кроме него, она еще способна проверять некоторые признаки документов на наличие подделок, а API позволяет строить многофункциональные решения, включающие в себя, например, сверку фотографии в документе с лицом перед камерой.

SmartID Reader состоит из нескольких крупных, относительно независимых блоков:

1. Поиск объекта на изображении. Эти функции призваны позволять находить на изображении документ (или говорить об его отсутствии). Для поиска других объектов они тоже используются: для поиска лиц или баркодов, например.
2. Выделение текста. Для дальнейшего распознавания после нахождения документа или же прямо сразу на входном изображении необходимо локализовать текстовые данные. В этом блоке происходит нахождение четырехугольников текста для дальнейшей обработки.
3. Распознавание строк. Функции данного блока принимают на вход изображение строки и в качестве ответа возвращают строку, где для каждого знакоместа указан не только символ, но и его оценкой вместе с альтернативными вариантами ответа и их оценками соответственно.
4. Постобработка результатов распознавания. Здесь задействованы разнообразные алгоритмы, позволяющие в настраиваемом режиме задать некоторые семантические правила, которым ответ обязан подчиняться. С учетом распределения оценок всех возможных символов строится наиболее правдоподобный результат. Эти правила могут включать в себя заданные вероятности специфических последовательностей символов, проверку словарей, автозамены и прочее.
5. Сравнение изображений объектов. Цель этого блока - уметь отвечать на вопрос: одно и то же ли изображено на представленных изображениях? Способ построения ответа может сильно отличаться в зависимости от постановки задачи, например, при сравнении флагов стран на документах и при

сравнении лиц на фотографиях должны использоваться совершенно разные модели.

6. Интеграция результатов. Так как система работает не с единичными изображениями, а с видеопотоком, то необходимо уметь объединять ответы работы различных подсистем с разных кадров в один ответ, отдаваемый пользователю. Так как ответы естественным образом могут весьма существенно изменяться от кадра к кадру, то здесь используется ряд нетривиальных алгоритмов, позволяющих, например, при некоторых обстоятельствах собрать правильный ответ даже в случае, когда все ответы с кадров по отдельности неверные.

Все нейронные сети, используемые в этой системе обучены и воспроизводятся с помощью разработанной системы обучения нейронных сетей и некоторые из них включают (или могут включать) слои прямого и транспонированного быстрого преобразования Хафа.

4.5 Результаты главы 4

В данной главе рассмотрены практические применения нейронных сетей, построенных с использованием представленных в главе 3 архитектур. В ходе этого были достигнуты следующие результаты:

1. Решена задача поиска внешних (включая несобственные) точек схода на изображении с использованием нейронной сети архитектуры Хафнет. Показано, что построенная end-to-end модель позволяет существенно улучшить качество в сравнении с инженерным подходом, использующим аналогичный алгоритм.
2. Решена задача семантической сегментации с помощью нейронной сети архитектуры Хафэнкодер, которая показала способность добиться большего качества, чем стандарт-

ная полносверточная сеть, имея на два порядка меньше обучаемых параметров. Благодаря анализу признаков в координатах (s, t) , нейронная сеть способна использовать более релевантные признаки, что позволяет значительно сократить в ней количество обучаемых параметров.

3. Решена задача поиска внутренней точки схода с помощью нейронной сети архитектуры Хафэнкодер. Продемонстрировано, что благодаря использованию интегральных признаков такая нейронная сеть способна решать задачу, даже если локальная область правильной точки на изображении заведомо испорчена. Это согласуется с предсказанием поведения нейронной сети в таком случае, основанном на элементах алгоритма поиска точек схода с помощью двух преобразований Хафа.
4. Решена задача поиска машиночитаемой зоны на изображении с помощью нейронной сети архитектуры Хафэнкодер. В этой задаче продемонстрировано, что, благодаря использованию интегральных признаков, сеть, имея на два порядка меньше коэффициентов, чем стандартная полносверточная, способна решить задачу с конкурентноспособным качеством.
5. Создан программный комплекс для обучения и воспроизведений нейронных сетей, включающих в себя эффективные реализации слоев прямого и транспонированного быстрого преобразования Хафа [94], а также получен патен США [95].

Заключение

В данной работе изучался вопрос использования интегральных операторов в качестве слоя нейронной сети. Для проведения всех экспериментов рассматривалось быстрое преобразование Хафа в качестве интегрального оператора. Для построения эффективных архитектур нейронных сетей с использованием такого оператора в качестве слоя было проведено детальное изучение быстрого преобразования Хафа, его имплементаций, а также транспонирования. Это позволило создать вычислительно эффективную имплементацию слоев прямого и транспонированного преобразования Хафа. В работе проведен анализ структуры рецептивного поля нейронной сети при наличии таких слоев в ней, из которого следует, что такие архитектуры будут иметь преимущества в определенном круге задач. Дополнительно в работе доказано, что выполнение транспонированного быстрого преобразования Хафа также возможно со сложностью $O(n^2 * \log(n))$, а не $O(n^3)$.

С использованием разработанных и изученных слоев было создано две основные нейросетевые архитектуры: Хафнет и Хафэнкодер, отвечающие разным задачам. Хафнет – архитектура с использованием двух последовательных слоев быстрого преобразования Хафа, разделенных блоками сверточных слоев. Благодаря своей архитектуре, координаты в карте признаков ответа такой сети связаны кусочно-проективным преобразованием с координатами исходного изображения, а бесконечно удаленные точки переходят в точки с конечными координатами. Это делает такую архитектуру особенно удобной для поиска несобственных точек схода. Вторая архитектура, Хафэнкодер, содержит в себе последовательное применение слоев прямого и транспонированного быстрого преобразования Хафа, разделенные блоками сверточных фильтров, что, в противоположность архитектуре Хафнет, позволяет получить результирующую карту признаков в координатах исходного изображения. Это удобно в случаях, когда, например, необходи-

мо найти объект, гарантированно лежащий внутри изображения. Обе архитектуры нейронных сетей для построения ответа способны учитывать интегральные признаки по всему изображению, а не в локальной окрестности, как это происходит для стандартных полносверточных сетей.

Эффективность разработанных нейросетевых архитектур была подтверждена различными экспериментами. Рассматривались задачи поиска несобственных точек схода на документах, где было продемонстрировано превосходство end-to-end схемы перед инженерным алгоритмом, задача семантической сегментации изображений документа, где было показано, что нейронная сеть архитектуры Хафэнкодер достигает лучшего качества в сравнении со стандартной архитектурой UNet, даже имея на два порядка меньше обучаемых параметров. Дополнительный эксперимент по поиску внутренней точки схода на изображениях дороги показал, что даже если “испортить” входные данные, закрыв саму точку схода, нейронная сеть архитектуры Хафэнкодер все равно способна ее найти, что полностью согласуется с выводами, сделанными при рассуждении о трансформации рецептивных полей.

В ходе работы над диссертацией были созданы имплементации, вошедшие в зарегистрированную систему обучения нейронных сетей, с помощью которой были обучены нейронные сети, встроенные в действующее коммерческое решение.

Конкретнее, в диссертации:

1. Показано, что добавление в сеть двух слоев преобразования Хафа последовательно, равно как и добавление последовательно слоев преобразования Хафа и транспонированного преобразования Хафа в многослойную сверточную сеть делает рецептивным полем каждого выходного нейрона все изображение.
2. Приведено доказательство возможности построения быстрого транспонированного оператора для быстрого преобразования Хафа, выполняемого со сложностью $O(n^2 * \log(n))$.

Это позволяет эффективно распространять градиент через слой преобразования Хафа при обучении.

3. Предложена архитектура Хафнет, содержащая в себе ветви с последовательно расположенными слоями быстрого преобразования Хафа, разделенными блоками сверточных слоев, а также метод ее применения, позволяющий кодировать на выходе всю плоскость входного изображения, включая бесконечно удаленные точки. Предложенным методом решена задача поиска внешних (включая несобственные) точек схода на изображении. Показано, что построенная end-to-end модель позволяет сократить ошибку на 20% в сравнении с инженерным подходом, использующим аналогичный алгоритм.
4. Предложена архитектура Хафэнкодер, содержащая в себе слои прямого и транспонированного быстрого преобразования Хафа, разделенные блоками сверточных слоев, что позволяет ей быть сетью типа “автокодировщик”. Предложенная архитектура в задаче семантической сегментации продемонстрировала лучшее качество (0.96 против 0.94 MIoU) при примерно в 100 раз меньшем числе обучаемых параметров, в сравнении со стандартной полносверточной сетью. Также решена задача поиска внутренней точки схода, где, в сравнении со стандартной нейронной сетью, ошибка оказалась примерно в 2 раза меньше во всех экспериментах. Продемонстрировано, что благодаря использованию интегральных признаков такая нейронная сеть способна решать задачу поиска точки схода, даже если в локальной окрестности искомой точки нет релевантных признаков.
5. Показано, что перенос обученных весов последних слоев сети Хафнет позволяет без потери качества применять ее к данным из другого домена (исследовались домены изображений документов и изображений с видеорегистратора автомобиля). Это демонстрирует, что перенос знаний в се-

тях таких архитектур возможен, как и в классических сетях, но происходит в глубоких слоях, работающих с признаками в пространстве Хафа, а не в первых слоях, как это происходит в ранее исследованных архитектурах.

6. Создан программный комплекс для обучения и воспроизведений нейронных сетей, включающих в себя эффективные реализации слоев прямого и транспонированного преобразования Хафа. Все практические результаты, приведенные в работе, были получены с его использованием.

Список сокращений и условных обозначений

Преобразование Радона – согласно определению с параметризацией (ρ, φ) . В преобразовании Радона: ρ – расстояние от центра изображения до прямой; φ – угол наклона прямой.

Преобразование Хафа – дискретное преобразование радона с параметризацией (ρ, φ) . В преобразовании Хафа: s – сдвиг первой точки прямой на стороне "начала"; t – сдвиг последней точки прямой на противоположной стороне (относительно начала).

Быстрое преобразование Хафа, БПХ – дискретное преобразование Радона с параметризацией (s, t) . Англоязычный аналог названия Fast Hough Transform, ФНТ.

Алгоритм Бредди – алгоритм вычисления быстрого преобразования Хафа, имеющий сложность $O(n^2 * \log(n))$, в отличие от наивного способа вычисления со сложностью $O(n^3)$.

Транспонированное быстрое преобразование Хафа – операция, получаемая в результате транспонирования матрицы оператора БПХ и ее применения. Используется как для осуществления обратного распространения градиента при обучении сетей со слоем БПХ, так и как самостоятельная операция для преобразования пространства из (s, t) в (x, y) . Англоязычный аналог названия Transposed Fast Hough Transform, ТФНТ.

Вектор признаков – вид данных, которыми оперируют слои нейронных сетей. Подразумевает, что элементы не зависят друг от друга.

Карта признаков – вид данных, которыми оперируют слои нейронной сети. Обычно имеются ввиду двумерные. Таким образом, соседние элементы имеют связь друг с другом, так как часто принадлежат одному объекту или структуре.

Полносвязный слой – вид слоя нейронной сети, в котором каждый элемент выхода является линейной комбинацией всех входов. Реализуется в виде умножения вектора входа на матрицу весов с добавлением вектора байесов.

Сверточный слой – вид слоя нейронной сети, где каждый элемент выходной карты признаков является линейной комбинацией элементов входной карты признаков, ограниченных окном (фильтром), расположенным в соответствующей позиции. В сверточном слое: f – размер окна фильтра; s – шаг приложения фильтра; n – количество фильтров; p – паддинг входного изображения; a – фактор увеличения размера изображения, деконволюции.

Слой пуллинга – вид слоя нейронной сети, применяемый к карте признаков. Операция похожа на применение сверточного фильтра, но вместо взвешенной суммы может быть среднее, минимум, максимум, медиана и т.д. Обычно используется операция максимума с целью понижения размерности карты признаков.

Паддинг – один из параметров сверточного слоя (может быть применен к слою пуллинга и к любым другим оконным слоям). Наличие паддинга означает, что перед применением фильтров картинка расширяется (или сужается в случае отрицательного паддинга). В зависимости от режима, в расширенной зоне могут быть нули, среднее по изображению, зеркальное отражение содержимого и т.п. Чаще всего используется паддинг нулями.

Функция активации – нелинейная функция, применяемая между линейными слоями нейронной сети. Самые часто используемые: ReLU $f(x) = \max(0, x)$, softsign $f(x) = x/(1 + |x|)$.

U-Net – архитектура полносверточной нейронной сети. Содержит в себе два ключевых элемента, наличие которых делает любую сеть U-Net-подобной. Первый: первая часть сети уменьшает размер карт признаков, увеличивая количество каналов, а вторая, наоборот, увеличивает размер, уменьшая количество каналов. Второй: добавление карт признаков к картам соответствующего размера, прошедшим через уменьшение-увеличение размера. Таким образом сеть способна не терять мелкие детали изображения.

Рецептивное поле нейрона – зона на исходной карте признаков, элементы которой могут влиять на значение в данном нейроне.

Точка схода – воображаемая точка на двухмерной проекции трехмерной сцены, в которой пересекаются прямые, параллель-

ные в исходной сцене. При работе с изображениями бывает двух видов. Внутренняя точка схода – точка схода, которая находится внутри рассматриваемого изображения. Внешняя точка схода – точка схода, которая находится за пределами рассматриваемого изображения. Частным случаем внешней точки схода является несобственная точка схода, которая находится на бесконечном удалении.

Полносверточная нейронная сеть – сеть без полносвязных слоев. Может быть применена к изображению произвольного размера. Размер выхода зависит от размера входа.

Список литературы

- [1] Tom M Mitchell и др. — «Machine learning». — В: (1997).
- [2] Nesreen Samer El_Jerjawi и Samy S Abu-Naser. — «Diabetes prediction using artificial neural network». — В: (2018).
- [3] T-L Lee. — «Neural network prediction of a storm surge». — В: *Ocean Engineering* 33.3-4 (2006), с. 483—494.
- [4] Moshe Leshno и Yishay Spector. — «Neural network prediction analysis: The bankruptcy case». — В: *Neurocomputing* 10.2 (1996), с. 125—147.
- [5] Swarnendu Ghosh и др. — «Understanding deep learning techniques for image segmentation». — В: *ACM Computing Surveys (CSUR)* 52.4 (2019), с. 1—35.
- [6] Pavel Vladimirovich Bezmaternykh, Dmitrii Alexeevich Ilin и Dmitry Petrovich Niko-
«U-Net-bin: hacking the document image binarization contest». — В: *Ком-
пьютерная оптика* 43.5 (2019).
- [7] Xian Wu, Kun Xu и Peter Hall. — «A survey of image synthesis and editing with generative adversarial networks». — В: *Tsinghua Science and Technology* 22.6 (2017), с. 660—674.
- [8] Tianyu Gu и др. — *Badnets: Evaluating backdooring attacks on deep neural networks*. — 2019.
- [9] Naveed Akhtar и Ajmal Mian. — «Threat of adversarial attacks on deep learning in computer vision: A survey». — В: *Ieee Access* 6 (2018), с. 14410—14430.
- [10] Vassileios Balntas и др. — «HPatches: A benchmark and evaluation of handcrafted and learned local descriptors». — В: *Proceedings of the IEEE conference on computer vision and pattern recognition*. — 2017, — С. 5173—5182.
- [11] Oludare Isaac Abiodun и др. — «State-of-the-art in artificial neural network applications: A survey». — В: *Heliyon* 4.11 (2018), e00938.
- [12] Ian Goodfellow и др. — *Deep learning*. — Т. 1. — 2. — MIT press Cambridge, 2016.

- [13] Frank Rosenblatt. — «The perceptron: a probabilistic model for information storage and organization in the brain.» — B: *Psychological review* 65.6 (1958), с. 386.
- [14] David E Rumelhart, Geoffrey E Hinton и Ronald J Williams. — «Learning representations by back-propagating errors». — B: *nature* 323.6088 (1986), с. 533—536.
- [15] Geoffrey E Hinton и Ruslan R Salakhutdinov. — «Reducing the dimensionality of data with neural networks». — B: *science* 313.5786 (2006), с. 504—507.
- [16] Yoshua Bengio. — «On the challenge of learning complex functions». — B: *Progress in Brain Research* 165 (2007), с. 521—534.
- [17] Alex Krizhevsky, Geoffrey Hinton и др. — «Learning multiple layers of features from tiny images». — B: (2009).
- [18] Filippo Biondi. — «Low-rank plus sparse decomposition and localized radon transform for ship-wake detection in synthetic aperture radar images». — B: *IEEE Geoscience and Remote Sensing Letters* 15.1 (2017), с. 117—121.
- [19] Guichao Lin и др. — «Fruit detection in natural environment using partial shape matching and probabilistic Hough transform». — B: *Precision Agriculture* 21.1 (2020), с. 160—177.
- [20] M Dian Bah, Adel Hafiane и Raphael Canals. — «Weeds detection in UAV imagery using SLIC and the hough transform». — B: *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*. — IEEE. 2017, — С. 1—6.
- [21] Xiaohan Yu и др. — «Radar moving target detection in clutter background via adaptive dual-threshold sparse Fourier transform». — B: *IEEE Access* 7 (2019), с. 58200—58211.
- [22] Y. Takezawa, M. Hasegawa и S. Tabbone. — «Camera-captured document image perspective distortion correction using vanishing point detection based on Radon transform». — B: *2016 23rd International Conference on Pattern Recognition (ICPR)*. — 2016, — С. 3968—3974. — DOI: 10.1109/ICPR.2016.7900254.
- [23] Ruohui Wang. — «Edge detection using convolutional neural network». — B: *International Symposium on Neural Networks*. — Springer. 2016, — С. 12—20.

- [24] Lovedeep Gondara. — «Medical image denoising using convolutional denoising autoencoders». — B: *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. — IEEE. 2016, — C. 241—246.
- [25] Seonhee Park и др. — «Dual autoencoder network for retinex-based low-light image enhancement». — B: *IEEE Access* 6 (2018), с. 22084—22093.
- [26] Michael van Ginkel, CL Luengo Hendriks и Lucas J van Vliet. — «A short introduction to the Radon and Hough transforms and how they relate to each other». — B: *Delft University of Technology* (2004).
- [27] Dana H Ballard. — «Generalizing the Hough transform to detect arbitrary shapes». — B: *Pattern recognition* 13.2 (1981), с. 111—122.
- [28] Jayanta Basak. — «Learning Hough transform: A neural network model». — B: *Neural computation* 13.3 (2001), с. 651—676.
- [29] M Köppen, A Soria-Frisch и R Vicente-Garciéa. — «NeuroHough: A neural network for computing the Hough transform». — B: *Artificial Neural Nets and Genetic Algorithms*. — Springer. 2001, — C. 197—200.
- [30] Martin L Brady и Whanki Yong. — «Fast parallel discrete approximation algorithms for the Radon transform». — B: *Proceedings of the fourth annual ACM symposium on Parallel algorithms and architectures*. — 1992, — C. 91—99.
- [31] E Ershov и др. — «Fast Hough transform analysis: pattern deviation from line segment». — B: *Eighth International Conference on Machine Vision (ICMV 2015)*. — Т. 9875. — International Society for Optics и Photonics. 2015, — C. 987509.
- [32] Nahum Kiryati и Alfred M Bruckstein. — «Heteroscedastic Hough transform (HtHT): An efficient method for robust line fitting in the ‘errors in the variables’ problem». — B: *Computer Vision and Image Understanding* 78.1 (2000), с. 69—83.
- [33] Jeremy Hsu. — «Biggest neural network ever pushes AI deep learning». — B: *IEEE Spectrum: Technology, Engineering, and Science News* (2015).
- [34] Jeff Rasley и др. — «Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters». — B: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. — 2020, — C. 3505—3506.

- [35] Florian Kluger и др. — «Deep learning for vanishing point detection using an inverse gnomonic projection». — В: *German Conference on Pattern Recognition*. — Springer. 2017, — С. 17—28.
- [36] Xue Li и др. — «Lane detection based on spiking neural network and hough transform». — В: *2015 8th International Congress on Image and Signal Processing (CISP)*. — IEEE. 2015, — С. 626—630.
- [37] Yichao Zhou и др. — «Neurvps: Neural vanishing point scanning via conic convolution». — В: *arXiv preprint arXiv:1910.06316* (2019).
- [38] Syed Ammar Abbas и Andrew Zisserman. — «A Geometric Approach to Obtain a Bird's Eye View From an Image.» — В: *ICCV Workshops*. — 2019, — С. 4095—4104.
- [39] Ali Borji. — «Vanishing point detection with convolutional neural networks». — В: *arXiv preprint arXiv:1609.00967* (2016).
- [40] Yongjie Shi и др. — «Three Orthogonal Vanishing Points Estimation in Structured Scenes Using Convolutional Neural Networks». — В: *2019 IEEE International Conference on Image Processing (ICIP)*. — IEEE. 2019, — С. 3537—3541.
- [41] Jian-Da Wu и Siou-Huan Ye. — «Driver identification using finger-vein patterns with Radon transform and neural network». — В: *Expert Systems with Applications* 36.3 (2009), с. 5793—5799.
- [42] Qutaishat Munib и др. — «American sign language (ASL) recognition based on Hough transform and neural networks». — В: *Expert systems with Applications* 32.1 (2007), с. 24—37.
- [43] Jinjiang Wang, Peilun Fu и Robert X Gao. — «Machine vision intelligence for product defect inspection based on deep learning and Hough transform». — В: *Journal of Manufacturing Systems* 51 (2019), с. 52—60.
- [44] Bo Liao и др. — «Hand gesture recognition with generalized hough transform and DC-CNN using realsense». — В: *2018 Eighth International Conference on Information Science and Technology (ICIST)*. — IEEE. 2018, — С. 84—90.
- [45] Kei-ichiro Minami, Hiroshi Nakajima и Takeshi Toyoshima. — «Real-time discrimination of ventricular tachyarrhythmia with Fourier-transform neural network». — В: *IEEE transactions on Biomedical Engineering* 46.2 (1999), с. 179—185.

- [46] Himanshu Gothwal, Silky Kedawat, Rajesh Kumar и др. — «Cardiac arrhythmias detection in an ECG beat signal using fast fourier transform and artificial neural network». — В: *Journal of Biomedical Science and Engineering* 4.04 (2011), с. 289.
- [47] Li-Hua Wang и др. — «Motor fault diagnosis based on short-time Fourier transform and convolutional neural network». — В: *Chinese Journal of Mechanical Engineering* 30.6 (2017), с. 1357—1368.
- [48] Harry Pratt и др. — «Fcnn: Fourier convolutional neural networks». — В: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. — Springer. 2017, — С. 786—798.
- [49] Christopher Syben и др. — «PYRO-NN: Python reconstruction operators in neural networks». — В: *Medical physics* 46.11 (2019), с. 5110—5115.
- [50] Bo Liang, SD Iwnicki и Yunshi Zhao. — «Application of power spectrum, cepstrum, higher order spectrum and neural network analyses for induction motor fault diagnosis». — В: *Mechanical Systems and Signal Processing* 39.1-2 (2013), с. 342—360.
- [51] Juan Ignacio Godino-Llorente и Pedro Gómez-Vilda. — «Automatic detection of voice impairments by means of short-term cepstral parameters and neural network based detectors». — В: *IEEE Transactions on Biomedical Engineering* 51.2 (2004), с. 380—384.
- [52] Erdem Yavuz и др. — «An epileptic seizure detection system based on cepstral analysis and generalized regression neural network». — В: *Biocybernetics and Biomedical Engineering* 38.2 (2018), с. 201—216.
- [53] Jan Adamowski и Hiu Fung Chan. — «A wavelet neural network conjunction model for groundwater level forecasting». — В: *Journal of Hydrology* 407.1-4 (2011), с. 28—40.
- [54] Xiaomo Jiang и Hojjat Adeli. — «Dynamic wavelet neural network model for traffic flow forecasting». — В: *Journal of transportation engineering* 131.10 (2005), с. 771—779.
- [55] Charles R Qi и др. — «Deep hough voting for 3d object detection in point clouds». — В: *Proceedings of the IEEE International Conference on Computer Vision*. — 2019, — С. 9277—9286.

- [56] Abol Basher и др. — «Hippocampus localization using a two-stage ensemble Hough convolutional neural network». — В: *IEEE Access* 7 (2019), с. 73436—73447.
- [57] Ming Yan и др. — «An end-to-end deep learning network for 3D object detection from RGB-D data based on hough voting». — В: *IEEE Access* 8 (2020), с. 138810—138822.
- [58] Thomas Wollmann и Karl Rohr. — «Deep residual Hough voting for mitotic cell detection in histopathology images». — В: *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*. — IEEE. 2017, — С. 341—344.
- [59] Fausto Milletari и др. — «Hough-CNN: deep learning for segmentation of deep brain regions in MRI and ultrasound». — В: *Computer Vision and Image Understanding* 164 (2017), с. 92—102.
- [60] А. В. Долматова и Д. П. Николаев. — «Ускорение свертки и обратного проецирования при реконструкции томографических изображений». — В: *Сенсорные системы* 34.1 (2020). Под ред. Михаил Аркадьевич. Островский, академик РАН и д.б.н. DOI: 10.31857/S0235009220010072, с. 64—71.
- [61] E Ershov и др. — «Fast Hough transform analysis: pattern deviation from line segment». — В: *Eighth International Conference on Machine Vision (ICMV 2015)*. — Т. 9875. — International Society for Optics и Photonics. 2015, — С. 987509.
- [62] Alexander Sheshkus и др. — «Vanishing point detection with direct and transposed fast Hough transform inside the neural network». — В: *Компьютерная оптика* 44.5 (2020).
- [63] Victor E Prun и др. — «A computationally efficient version of the algebraic method for computer tomography». — В: *Automation and remote control* 74.10 (2013), с. 1670—1678.
- [64] Alexander Sheshkus и др. — «Combining convolutional neural networks and hough transform for classification of images containing lines». — В: *Ninth International Conference on Machine Vision (ICMV 2016)*. — Т. 10341. — International Society for Optics и Photonics. 2017, — С. 103411C.

- [65] А. В. Шешкус. — «Использование сверточных нейронных сетей в комбинации с преобразованием Хафа для классификации изображений с прямыми линиями». — В: *Труды ИСА РАН* 67.1 (2017). Под ред. Попков Юрий Соломонович и др., с. 83—88.
- [66] E. Kuznetsova, E. Shvets и D. Nikolaev. — «Viola-Jones based hybrid framework for real-time object detection in multispectral images». — В: *ICMV 2015*. — Под ред. Antanas Verikas; Petia Radeva; Dmitry Nikolaev. — Т. 9875. — DOI: 10.1117/12.2228707. — Bellingham, Washington 98227-0010 USA: Society of Photo-Optical Instrumentation Engineers (SPIE), дек. 2015, — 98750N1—98750N6.
- [67] Vincent Vanhoucke, Andrew Senior и Mark Z Mao. — «Improving the speed of neural networks on CPUs». — В: (2011).
- [68] Roberto Rigamonti и др. — «Learning separable filters». — В: *Proceedings of the IEEE conference on computer vision and pattern recognition*. — 2013, — С. 2754—2761.
- [69] Elena Limonova, Dmitry Ilin и Dmitry Nikolaev. — «Improving Neural Network Performance on SIMD Architectures». — В: *ICMV 2015*. — Под ред. Antanas Verikas; Petia Radeva; Dmitry Nikolaev. — Т. 9875. — DOI: 10.1117/12.2228594. — Bellingham, Washington 98227-0010 USA: Society of Photo-Optical Instrumentation Engineers (SPIE), дек. 2015, — С. 98750L1—98750L6.
- [70] Elena Limonova, Alexander Sheshkus и Dmitry Nikolaev. — «Computational optimization of convolutional neural networks using separated filters architecture». — В: *IJAER* 11.11 (2016). Под ред. Editor in Chief: Prof. Ir Dr Mohd с. 7491—7494.
- [71] Yuliya Aleksandrovna Shemyakina и др. — «A method of image quality assessment for text recognition on camera-captured and projectively distorted documents». — В: *Mathematics* 9.17 (2021). DOI: 10.3390/math9172155, с. 1—22.
- [72] Aleksandr Sheshkus, Anastasiya Ingacheva и Dmitry Nikolaev. — «Vanishing Points Detection Using Combination of Fast Hough Transform and Deep Learning». — В: *ICMV 2017*. — Под ред. Antanas Verikas и др. — Т. 10696. — DOI: 10.1117/12.2310170. — Bellingham, Washington 98227-0010

- USA: Society of Photo-Optical Instrumentation Engineers (SPIE), апр. 2018, — 106960H1—106960H8.
- [73] Alexander Sheshkus и др. — «HoughNet: neural network architecture for vanishing points detection». — В: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. — IEEE. 2019, — С. 844—849.
 - [74] Alexander Sheshkus, Dmitry Nikolaev и Vladimir L Arlazarov. — «Houghencoder: Neural Network Architecture for Document Image Semantic Segmentation». — В: *2020 IEEE International Conference on Image Processing (ICIP)*. — IEEE. 2020, — С. 1946—1950.
 - [75] Hyeonwoo Noh, Seunghoon Hong и Bohyung Han. — «Learning deconvolution network for semantic segmentation». — В: *Proceedings of the IEEE international conference on computer vision*. — 2015, — С. 1520—1528.
 - [76] Hongyang Gao и др. — «Pixel transposed convolutional networks». — В: *IEEE transactions on pattern analysis and machine intelligence* 42.5 (2019), с. 1218—1227.
 - [77] Alexander V Sheshkus и Dmitry Nikolaev. — «Transfer of a high-level knowledge in HoughNet neural network». — В: *Twelfth International Conference on Machine Vision (ICMV 2019)*. — Т. 11433. — International Society for Optics и Photonics. 2020, — С. 1143322.
 - [78] Alexander Sheshkus, Anastasia Ingacheva и Dmitry Nikolaev. — «Vanishing points detection using combination of fast Hough transform and deep learning». — В: *Tenth International Conference on Machine Vision (ICMV 2017)*. — Т. 10696. — International Society for Optics и Photonics. 2018, — 106960H.
 - [79] *High-performance C++/CUDA implementation of convolutional neural networks, <https://code.google.com/p/cuda-convnet/>.*
 - [80] V. V. Arlazarov и др. — «MIDV-500: A Dataset for Identity Document Analysis and Recognition on Mobile Devices in Video Stream». — В: *Computer Optics* 43.5 (2019). Под ред. Soyfer Viktor Aleksandrovich. DOI: 10.18287/2412-6179-2019-43-5-818-824, с. 818—824.

- [81] Haikal El Abed, Liu Wenying и Volker Margner. — «International conference on document analysis and recognition (ICDAR 2011)-competitions overview». — В: *2011 International Conference on Document Analysis and Recognition*. — IEEE. 2011, — С. 1437—1443.
- [82] Ray Smith. — «An overview of the Tesseract OCR engine». — В: *Ninth international conference on document analysis and recognition (ICDAR 2007)*. — Т. 2. — IEEE. 2007, — С. 629—633.
- [83] Yusuke Takezawa, Makoto Hasegawa и Salvatore Tabbone. — «Camera-captured document image perspective distortion correction using vanishing point detection based on radon transform». — В: *2016 23rd International Conference on Pattern Recognition (ICPR)*. — IEEE. 2016, — С. 3968—3974.
- [84] Yusuke Takezawa, Makoto Hasegawa и Salvatore Tabbone. — «Robust perspective rectification of camera-captured document images». — В: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. — Т. 6. — IEEE. 2017, — С. 27—32.
- [85] Konstantin Bulatov, Daniil Matalov и Vladimir V. Arlazarov. — «MIDV-2019: Challenges of the Modern Mobile-Based Document OCR». — В: *ICMV 2019*. — Под ред. Wolfgang Osten, Dmitry Nikolaev и Jianhong Zhou. — Т. 11433. — DOI: 10.1117/12.2558438. — Bellingham, Washington 98227-0010 USA: Society of Photo-Optical Instrumentation Engineers (SPIE), янв. 2020, — 114332N1—114332N6.
- [86] А. В. Гайер, А. В. Шешкус и Ю. С. Чернышова. — «Аугментация обучающей выборки "на лету" для обучения нейронных сетей». — В: *Труды ИСА РАН* 68. Спецвыпуск № S1 (2018). Под ред. Попков Юрий Соломонович и др. DOI: 10.14357/20790279180517, с. 150—157.
- [87] S. Ilyuhin и др. — «Hough Encoder for Machine Readable Zone Localization». — В: *Pattern Recognit. Image Anal.* 32.4 (2022). Под ред. Zhuravlev Y. I. DOI: 10.1134/S1054661822040150, с. 793—802.
- [88] Andreas Hartl, Clemens Arth и Dieter Schmalstieg. — «Real-time Detection and Recognition of Machine-Readable Zones with Mobile Devices». — В: *VISAPP*. — 2015.

- [89] SI Kolmakov, NS Skoryukina и VV Arlazarov. — «Machine-Readable Zones Detection in Images Captured by Mobile Devices' Cameras». — В: *Pattern Recognition and Image Analysis* 30.3 (2020), с. 489—495.
- [90] NS Skoryukina. — «Machine-readable zones localization method robust to capture conditions». — В: *Trudy ISA RAN* 67.4 (2017), с. 80—85.
- [91] Yulia S. Chernyshova, Alexander V. Sheshkus и Vladimir V. Arlazarov. — «Two-step CNN framework for text line recognition in camera-captured images». — В: *IEEE Access* 8 (2020). Под ред. Prof. Derek Abbott. DOI: 10.1109/ACCESS.2020.2974051, с. 32587—32600.
- [92] Aleksandr Vyacheslavovich Gayer, Darya Mikhaylovna Ershova и Vladimir Viktorovich Arlazarov. — «Fast and accurate deep learning model for stamps detection for embedded devices». — В: — Под ред. Zhuravlev Y. I. — Т. 32. — 4. — DOI: 10.1134/S1054661822040046. — Road Town, Tortola, British Virgin Islands, United Kingdom: Pleiades Publishing, Ltd., 2022, — С. 772—779.
- [93] A. V. Yamaev и др. — «Neural network regularization in the problem of few-view computed tomography». — В: *Computer Optics* 46.3 (2022). Под ред. Soyfer Viktor Aleksandrovich. DOI: 10.18287/2412-6179-CO-1035, с. 422—428.
- [94] Александр Владимирович Шешкус и др. — *Автоматическая система генерации данных и обучения искусственных нейронных сетей "Smart NNCreator"*. — № 2018615794. — Бережковская наб., д. 24, стр. 12, Москва, Россия., 2018.
- [95] Alexander Vladimirovich SHESHKUS и др. — *Artificial Intelligence Using Convolutional Neural Network With Hough Transform*. — Pub. No.: US 2022/0122267 A1; Patent No.: 17 / 237539. — 2022.

Список рисунков

1.1	Две из трех волн развития нейронных сетей.	17
1.2	Применение ядра сверточного фильтра размером 2×2 к многоканальной карте признаков.	18
1.3	Вычисление преобразования Радона от функции двух переменных.	22
1.4	Несобственные (слева) и собственная (справа) точки схода.	23
1.5	Быстрое преобразование Хафа от изображения с единственной точкой. Схематичное изображение.	25
1.6	Схема обратного гномонического проецирования.	27
2.1	Образование рецептивного окна в полносверточной сети.	36
2.2	Одномерная функция с двумя единичными точками (слева) и ее интеграл (справа).	37
2.3	Образование рецептивного окна при наличии одного слоя быстрого преобразования Хафа.	38
2.4	Диапазоны углов для вычисления БПХ по четвертям. .	40
2.5	Переход прямой в точку при быстром преобразовании Хафа.	41
2.6	Исходное изображение (слева), БПХ без сдвига (в центре), БПХ со сдвигом (справа).	43
2.7	Образование “расчески” при сдвиге пикселизированной прямой.	43
2.8	Все возможные паттерны для длины 4.	45
2.9	Матрица смещения паттернов	48
2.10	Матрица БПХ, исходная слева, преобразованная справа.	50
2.11	Приращение угла нелинейно зависит от сдвига.	53
2.12	Примеры изображений и результатов БПХ для них. . .	54

3.1	Архитектура Хафэнкодера в общих чертах. Блоки 1 и 7 - вход и выход соответственно. Блок 2 - сверточные слои, работающие с исходным изображением. Блоки 3 и 5 слои прямого и транспонированного БПХ. Блок 4 - сверточные слои, работающие в координатах (s, t) . Блок 6 - сверточный блок, работающий в координатах (x, y) , но уже с глобальными признаками.	66
3.2	Трансформация признаков в Хафэнкодере и в UNet. . .	68
3.3	Примеры контуров для экспериментов.	69
3.4	Примеры идеалов для контуров с изображения 3.3. . . .	69
3.5	Ответы полносверточной сети для изображений 3.3. . .	70
3.6	Ответы Хафэнкодера для изображений 3.3.	70
3.7	Способ перевода собственной точки схода в несобственную для изображений дорог.	73
3.8	Примеры изображений с соответствующей разметкой для документов (сверху) и для дорог (снизу).	73
4.1	Стандартная и модифицированная архитектуры.	78
4.2	Примеры из нейронной сети: а) Входное изображение; б) Идеальное изображение; в) БПХ идеального изображения; г) БПХ исходного изображения; д) БПХ исходного изображения после применения обученных сверточных слоев.	79
4.3	Примеры из набора данных MIDV-500.	82
4.4	Примеры изображений из набора данных ICDAR 2011. .	83
4.5	Процесс выправления документа. Сверху – ICDAR 2011, снизу – MIDV-500.	85
4.6	Сравнение работы Хафэнкодера и UNet в сложных случаях. Входное изображение (слева), результат UNet (в центре), результат Хафэнкодера (справа).	87
4.7	Примеры изображений документов с машиночитемой зоной.	88

4.8	Извлечение изображения машичитаемой зоны: исходное изображение, карта семантической сегментации, наложенная на изображение карта, зоны строк, машиночитаемая зона, извлеченное изображение машиночитаемой зоны.	90
4.9	Примеры изображений.	92
4.10	Примеры карт признаков с разных слоев нейронной сети: а) вход сети; б) выход первого блока сверточных слоев; с) выход прямого БПХ; d) вход транспонированного БПХ; e) выход транспонированного БПХ; f) выход сети.	93
4.11	Примеры добавленных в изображения искажений, призванных сделать невозможным локальный поиск точки схода.	94
4.12	Деградация качества при порче изображения для решеток различного размера.	95

Список таблиц

1	Точность различных нейросетевых архитектур при наличии и отсутствии результатов преобразования Хафа во входе второй ветви на наборах данных CIFAR-10 и строках из русского паспорта. N_{source} – количество фильтров в ветви для исходного изображения, N_{hough} – количество фильтров в ветви для преобразования Хафа.	56
2	Архитектура Хафнет	63
3	Архитектура нейронной сети Хафэнкодер	67
4	Сравнение ошибки в трех экспериментах	74
5	Результаты на наборе данных MIDV-500.	83
6	Результаты на наборе данных ICDAR 2011.	84
7	Сравнительные результаты, MIoU	87
8	Сравнительные результаты распознавания машиночитаемой зоны при разных методах поиска. . . .	91
9	Сравнение ошибок представленного метода с базовым .	94

Приложение А

Акты о внедрении

Акт о внедрении результатов диссертационной работы в деятельность ООО «Смарт Энджинс Сервис».



ООО Смарт Энджинс Сервис 117312, город Москва,
ОГРН: 1167746085297 пр-кт 60-Летия Октября, д. 9
ИНН: 7728328449

T: +7 (495) 649-82-60
E: office@smartengines.ru
https://smartengines.ru

17.02.2023 № 004

На № от

АКТ

об использовании (внедрении) результатов диссертационной работы Шешкуса Александра Владимировича «Использование преобразования Хафа в качестве слоя нейронной сети» в программных продуктах ООО «Смарт Энджинс Сервис»

Предложенные А. В. Шешкусом слои для нейронных сетей и использующие их архитектуры позволили повысить качество и скорость детекции различных объектов, требующихся для построения системы распознавания документов, удостоверяющих личность, в видеопотоке мобильных устройств и на последовательностях кадров, получаемых со стационарных камер. Результаты диссертации Шешкуса Александра Владимировича внедрены в программы для ЭВМ «Smart NNCreator» (свидетельство о государственной регистрации программы для ЭВМ № 2018615704 от 15.05.2018) и «Smart CardReader» (свидетельство о государственной регистрации программы для ЭВМ № 2016616615 от 16.06.2016).

Данные программы используются следующими организациями:

- ФНС РФ, МВД РФ, НСПК «МИР», государственных информационных системах Федерального дорожного агентства «Росавтодор», Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации (паспортно-визовая система ГС Мир);
- информационных системах крупных российских и зарубежных финансовых организаций, таких как ПАО Банк ВТБ, АО «Почта Банк», АО «Тинькофф Банк», АО «АльфаСтрахование, АО «АЛЬФА-БАНК», ПАО Банк «ФК Открытие», АО «Газпромбанк», Евразийский банк развития, PJSC Emirates NBD Bank;
- сотовых операторов ПАО «МТС», «МегаФон», «ВымпелКом» (торговая марка «Билайн»);
- в составе автоматизированной системы паспортного контроля «Сапсан», предназначенной для осуществления автоматического паспортного контроля пассажиров, следующих через государственную границу Российской Федерации;
- системах автоматизации продажи билетов ОАО «РЖВ», а также авиакомпаний (Turkish Airlines, Croatia airlines).

Генеральный директор
ООО «Смарт Энджинс Сервис»
Арлазаров Владимир Викторович



Акт о внедрении результатов диссертационной работы в деятельность ПАО «ВТБ».



АКТ

**об использовании (внедрении) результатов диссертационной работы Шешкуса
Александра Владимировича «Использование преобразования Хафа в качестве слоя
нейронной сети» в Банке ВТБ**

Результаты диссертационной работы «Использование преобразования Хафа в качестве слоя нейронной сети» обладают высокой актуальностью и представляют практический интерес для решения задач поиска баркодов, банковских карт и документов на изображениях, полученных с мобильных устройств.

Технологии распознавания в видеопотоке на мобильных устройствах, использующие разработанные А.В. Шешкусом методы, позволяют повысить качество работы, скорость и безопасность распознавания и, таким образом, улучшить качество и эффективность обслуживания клиентов в банковской сфере. Данные технологии в составе программных продуктов ООО «Смарт Энджинс Сервис» внедрены и используются в веб-приложении и мобильных приложениях Банка ВТБ (ПАО).

С уважением,

Начальник Управления Платежи

Департамента транзакционного розничного бизнеса

A handwritten signature in black ink, appearing to be 'R.V. Novikov', written over a light blue circular stamp.

Р.В. Новиков

Акт о внедрении результатов диссертационной работы в информационные системы и мобильные приложения АО «Тинькофф Банк».



ТИНЬКОФФ

АКЦИОНЕРНОЕ ОБЩЕСТВО «ТИНЬКОФФ БАНК»
РОССИЯ, 127287, МОСКВА, УЛ. 3-Я ХУТОРСКАЯ, Д. 38А, СТР. 26
ТЕЛ.: +7 495 648-10-00, TINKOFF.RU

Исх. № КБ-0217.44
От 17.02.2023 г.

АКТ

об использовании (внедрении) результатов диссертационной работы Шешкуса Александра Владимировича «Использование преобразования Хафа в качестве слоя нейронной сети» в АО «Тинькофф Банк»

Результаты диссертационной работы «Использование преобразования Хафа в качестве слоя нейронной сети» обладают высокой актуальностью и представляют практический интерес для решения задач поиска банковских карт и идентификационных документов на изображениях, полученных с мобильных устройств.

Технологии распознавания в видеопотоке на мобильных устройствах, использующие разработанные А.В. Шешкусом методы, позволяют повысить качество работы, скорость и безопасность распознавания документов и, таким образом, улучшить качество и эффективность обслуживания клиентов в банковской сфере. Данные технологии в составе программных продуктов ООО «Смарт Энджинс Сервис» внедрены и используются в информационных системах и мобильных приложениях АО «Тинькофф Банк».

«Тиньков Банк» был признан лучшим розничным онлайн-банком в мире в 2020 и 2018 гг. по версии Global Finance. В 2020 г. Банк также стал победителем в категории «Лучший розничный европейский банк» международной банковской премии Retail Banker International Awards. Мобильное приложение банка регулярно признается лучшим на рынке российским и международными независимыми экспертами (Deloitte в 2013, 2014, 2015 и 2016 гг., Global Finance в 2018 г.).

Директор по информационным технологиям,
Заместитель председателя правления
АО «Тинькофф Банк»

В.В. Цыганов