

Федеральное государственное бюджетное образовательное  
учреждение высшего профессионального образования  
«Тульский государственный университет»  
Факультет Кибернетики  
Кафедра автоматики и телемеханики

На правах рукописи

Динь Вьет Шанг

**Алгоритмы подбора параметров комбинирования  
ациклических графов соседства в задачах обработки  
текстурных изображений**

05.13.17 – Теоретические основы информатики

диссертация на соискание ученой степени  
кандидата технических наук

Тула, 2013

## ОГЛАВЛЕНИЕ

<b>ВВЕДЕНИЕ.....</b>	<b>4</b>
<b>1 ЗАДАЧА РАСПОЗНАВАНИЯ ОБРАЗОВ В МАССИВАХ ВЗАИМОСВЯЗАННЫХ ДАННЫХ.....</b>	<b>8</b>
1.1 Классическая задача распознавания образов.....	8
1.2 Задача распознавания в массивах взаимосвязанных данных и минимизация среднего риска .....	10
1.3 Распознавание образов в массивах взаимосвязанных данных в случае ациклического графа соседства .....	19
1.4 Частная модель скрытого марковского случайного поля принадлежностей объектов к классам .....	22
1.5 Итерационный алгоритм повторения ациклического графа и комбинирование ациклических графов соседства .....	26
1.6 Натуральные и структурные параметры модели в задаче выбора модели ..	31
1.7 Минимизация гиббсовской энергии взаимодействия элементов марковского случайного поля .....	35
1.8 Основные цели и задачи исследования.....	40
<b>2 ЗАДАЧА ПОДБОРА ПАРАМЕТРОВ ДРЕВОВИДНОГО МАРКОВСКОГО СЛУЧАЙНОГО ПОЛЯ .....</b>	<b>42</b>
2.1 Постановка задачи подбора диагонального элемента в случае ациклического графа соседства .....	42
2.2 Предельные значения диагонального элемента .....	47
2.3 Алгоритм подбора диагонального элемента для ациклического графа, основанный на независимом обучении.....	54
2.5 Алгоритм подбора диагонального элемента для ациклического графа по схеме Гаусса-Зайделя .....	55
<b>3 ЗАДАЧА ПОДБОРА ПАРАМЕТРОВ КОМБИНИРОВАНИЯ АЦИКЛИЧЕСКИХ ГРАФОВ СОСЕДСТВА.....</b>	<b>57</b>
3.1 Свойства алгоритма подбора весов ациклических графов в их линейной комбинации.....	57
3.1.1 Предварительные эксперименты в случае однократного распознавания.....	59
3.1.2 Предварительные эксперименты в случае многократного распознавания.....	64
3.2 Алгоритмы подбора параметров комбинирования ациклических графов соседства.....	68

3.2.1 Алгоритм подбора единственного диагонального элемента и весов ациклических графов соседства.....	68
3.2.2 Первая схема подбора диагональных элементов и весов ациклических графов соседства.....	69
3.2.3 Вторая схема подбора диагональных элементов и весов ациклических графов соседства.....	71
3.2.4 Сходимость алгоритмов подбора параметров комбинирования ациклических графов соседства.....	72
<b>4 ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ АЛГОРИТМОВ .....</b>	<b>73</b>
4.1 Модельные текстурные изображения .....	73
4.2 Сравнение алгоритмов подбора диагонального элемента для ациклического графа .....	75
4.3 Зависимость скорости сходимости от значения диагонального элемента....	84
4.4 Сравнение алгоритмов подбора параметров комбинирования ациклических графов соседства.....	85
4.5 Проверка статистической гипотезы о равенстве средних уровня ошибок распознавания.....	100
4.6 Распознавание неподходящих изображений.....	102
<b>5 ОЦЕНКА ОШИБКИ РАСПОЗНАВАНИЯ МЕТОДОМ СКОЛЬЗЯЩЕГО КОНТРОЛЯ .....</b>	<b>104</b>
5.1 Упрощенная схема скользящего контроля.....	104
5.2 Алгоритмы подбора диагонального элемента по ациклическому графу.....	107
5.3 Алгоритмы подбора параметров комбинирования ациклических графов соседства.....	108
<b>6 РАСПОЗНАВАНИЕ РЕАЛЬНЫХ ИЗОБРАЖЕНИЙ .....</b>	<b>110</b>
6.1 Интерактивная схема выбора данных учителя.....	110
6.2 Схема независимого обучения с использованием информации о расположении объектов классов в поле носителя .....	116
6.3 Сравнение алгоритмов при распознавании реальных изображении .....	118
6.4 Обновление апостериорных маргинальных распределений.....	120
6.5 Обработка изображений известных баз.....	139
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>145</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....</b>	<b>148</b>
<b>ПРИЛОЖЕНИЕ .....</b>	<b>153</b>

## ВВЕДЕНИЕ

В настоящее время сохраняется большой интерес исследователей к задачам компьютерной обработки массивов данных, таких как сигналов, двумерных изображений, а также многомерных кубов данных.

Изображения являются одним из наиболее распространенных видов информации, ставшим в последние десятилетия типовым объектом применения компьютеров для анализа данных. Широко известны системы распознавания отпечатков пальцев, анализа электрокардиограмм, а что касается программ автоматического чтения печатного текста, то их использование стало массовым.

Особый интерес к компьютерной обработке именно изображений в значительной мере определяется тем фактом, что это естественный вид организации потоков информации о внешнем мире, получаемой человеком и другими высшими животными через органы чувств, главным образом, посредством зрения, и играющей фундаментальную роль в формировании их поведения.

Изображение не может быть представлено в компьютере иначе, как совокупность чисел, и эта совокупность упорядочена вдоль осей двух переменных, либо даже трех, если анализу подлежит изображение, изменяющееся во времени, как это, вообще говоря, и имеет место в зрительных системах человека и животных.

Одной из известных задач обработки изображений является задача сегментации изображений [26] – разделение плоскости изображения на непесекающиеся области, которые в некотором смысле однородны.

В данной задаче необходимо принять решение о принадлежности каждой точки изображения к тому или иному классу текстуры. В классической теории распознавания образов [2, 16, 26] объекты, подлежащие распознаванию, рассматриваются независимо друг от друга. Поэтому в классической задаче распознавания, в частности, не делается никаких предположений о порядке предъявления объектов. Однако во многих задачах объектами, под-

лежащими распознаванию, являются, например, моменты изменения характеристик некоторого развивающегося во времени процесса или характеристики локальных участков некоторой распределенной в пространстве среды. В таких задачах множество всех объектов, подлежащих распознаванию, составляет некоторый единый массив, обусловленный природой исследуемого явления – его протяженностью во времени или в пространстве. К объектам такого массива очевидным образом применимы понятия «смежности», «соседства», «упорядоченности». В частности, это приводит к тому, что порядок предъявления объектов становится чрезвычайно важным. Как следствие, возникает новое свойство распознаваемой совокупности – взаимосвязанность составляющих ее объектов, которую довольно часто представляют неориентированным графом без петель, который называют графом соседства [7, 8].

Применение скрытых марковских моделей для изучения зависимых наблюдений показало их высокую эффективность при обработке массивов линейно упорядоченных объектов с цепочечной смежностью их элементов [15, 42]. Однако применение данной идеи для отношений соседства произвольного вида выявило существенные теоретические и практически трудности. Для графов соседства общего вида, содержащих, как правило, циклы, задача распознавания марковских случайных полей является весьма трудоемкой [31, 37, 45, 49] и обладает свойствами задачи класса  $NP$ . Поэтому в интеллектуальном анализе данных и машинном обучении в настоящее время интенсивно развивается область, получившая название графических моделей [21, 22, 32, 36, 37, 49–52] которые опираются на графы соседства элементов множества для построения эффективных алгоритмов обработки данных, в том числе и для изображений. В частности, если граф соседства является ациклическим, то ранее был предложен базовый алгоритм, выполняемый всего за два или три прохода по ациклическому графу [7, 8, 9].

Очевидно, что произвольный граф соседства  $G$  нельзя заменить деревом без потери его фундаментального свойства нести полную информацию о положении каждого элемента массива относительно всех его других элемен-

тов. Чтобы уменьшить потери, связанные с древовидной аппроксимацией исходного графа соседства, в предыдущих работах был предложен алгоритм комбинирования ациклических графов, каждый из которых обладает коэффициентом важности, называемым его весом [11, 12]. Однако оказывается, что разные наборы весов ациклических графов дадут разные результаты распознавания. Для нахождения оптимальных весов графов был разработан алгоритм определения весов графов [12].

С целью упрощения модели ранее также была предложена частная модель марковского случайного поля, выраженная через модель марковской цепи. Такая модель случайного поля определяется единой матрицей условных вероятностей переходов, которая задается одним значением ее диагонального элемента.

Однако это значение ранее задавалось эвристически без поиска его оптимального значения диагонального элемента.

В данной работе предложены алгоритмы подбора диагонального элемента для заданного ациклического графа и алгоритмы подбора параметров комбинирования ациклических графов соседства, т.е. алгоритмы подбора весов графов, дополненные подбором значения диагонального элемента. Цель данной работы – это разработка алгоритмов подбора диагонального элемента для ациклического графа и включение поиска значения диагонального элемента в схему Гаусса-Зайделя [3] при подборе весов графов.

Для достижения указанной цели в данной работе поставлены следующие задачи:

- сформулировать задачу подбора диагонального элемента;
- разработать алгоритмы подбора диагонального элемента для заданного ациклического графа;
- сформулировать задачу подбора параметров комбинирования ациклических графов соседства;
- разработать алгоритмы подбора параметров комбинирования ациклических графов соседства;

- исследовать разработанные алгоритмы в задаче сегментации текстурных изображений.

Данная работа состоит из введения, шести глав и заключения.

В *первой главе* рассмотрена классическая задача распознавания образов и задача обработки взаимосвязанных массивов данных. Рассматривается базовый алгоритм распознавания образов в массивах взаимосвязанных данных в случае ациклического графа соседства, и описывается частная модель скрытого марковского поля принадлежностей объектов к классам. Рассмотрена задача настройки структурных параметров. Рассмотрена задача гиббсовской минимизации функции энергий связей. Формулируются основные цели и задачи исследования.

Во *второй главе* дается постановка задачи поиска диагонального элемента. Рассматриваются предельные значения диагонального элемента. Разрабатываются алгоритмы подбора диагонального элемента для ациклического графа соседства.

В *третьей главе* дается постановка задачи подбора параметров комбинирования ациклических графов соседства, приводятся предварительные эксперименты для исследования свойств предложенного ранее алгоритма подбора весов и разрабатываются алгоритмы подбора параметров комбинирования ациклических графов соседства.

В *четвертой главе* описывается экспериментальное исследование по распознаванию трех классов текстур тестовых текстурных изображений с помощью разработанных алгоритмов и других предложенных ранее алгоритмов. Проводится анализ полученных результатов и на их основе делаются выводы о разработанных в данной работе алгоритмах.

В *пятой главе* выполнены эксперименты по упрощенной схеме скользящего контроля [19], чтобы проверить качество решающего правила, построенное предложенными алгоритмами.

В *шестой главе* выполнены эксперименты по обработке реальных изображений.

# 1 ЗАДАЧА РАСПОЗНАВАНИЯ ОБРАЗОВ В МАССИВАХ ВЗАИМОСВЯЗАННЫХ ДАННЫХ

## 1.1 Классическая задача распознавания образов

Способность «распознавать» считается основным свойством человеческих существ, как, впрочем, и других живых организмов. В каждое мгновение нашего бодрствования мы совершаем акты распознавания. Мы опознаем окружающие нас объекты и в соответствии с этим перемещаемся и совершаем определенные действия.

Согласно [16], распознавание человеком конкретных образов можно рассматривать как психофизическую задачу, связанную с процессом взаимодействия индивида с определенным физическим раздражителем. В сущности, распознавание человеком образов можно свести к вопросу оценки относительных шансов на то, что исходные данные соответствуют совокупностям, определяемым прошлым опытом человека и предоставляющим ориентиры и априорную информацию для распознавания. Таким образом, задачу распознавания образов можно рассматривать как задачу установления различий между исходными данными, причем не посредством отождествления с отдельными образами, но с их совокупностями. Распознавание образов можно определить как отнесение исходных данных к определенному классу с помощью выделения существенных признаков или свойств, характеризующих эти данные, из общей массы несущественных деталей.

Образ – это описание любого объекта как представителя соответствующего класса образов [16]. Наиболее удобным математическим описанием объектов считается векторное описание. В этом случае каждому объекту  $t$  ставится в соответствие некоторый вектор признаков  $y_t$  этого объекта, который является элементом векторного пространства  $y_t \in \Theta$ , определенного природой источника данных. Такое векторное пространство  $\Theta$  называется пространством признаков. Как правило, это пространство является конечномерным и метрическим. Если признаки такого пространства являются вещественными величинами, то такое пространство изоморфно метрическому



пространству  $\mathbf{R}^n$ , где  $n$  – размерность пространства признаков. Тогда вектор признаков имеет вид  $\mathbf{y}_t = (y_1, y_2, \dots, y_n)^T$  [2]. Выбор наиболее информативных признаков – это одна из основных и важных задач в теории распознавания образов [2]. Однако в данной работе эта проблема не рассматривается.

В [2] была введена математическая постановка задачи распознавания образов. Пусть  $T$  – множество объектов. Отдельный объект из этого множества будем обозначать символом  $t$ . Каждый объект обладает своим вектором признаков  $\mathbf{y}_t$ , принимающим значения из пространства признаков  $\Theta$ , которое, чаще всего, является конечномерным и метрическим. Пусть  $P: T \rightarrow \Theta$  – оператор, отображающий объект  $t \in T$  в вектор признаков  $\mathbf{y}_t \in \Theta$ .

Предполагается, что на множестве объектов  $T$  в данной задаче распознавания нас интересуют некоторые подмножества – классы. В классической задаче распознавания образов считается, что множество классов является конечным, и классы образуют полную группу подмножеств из множества объектов  $T$ . В классической задаче рассматриваются обособленные объекты, каждый из которых объективно принадлежит к одному классу, и предъявляется для распознавания независимо от других объектов [8, 9]. В общем случае классов может быть и бесконечно много, и они могут составлять полную группу множеств. Задачу распознавания образов в этом случае называют обобщенной [2], и в данной работе она не рассматривается.

Пусть  $\Omega = \{\omega_1, \omega_2, \dots, \omega_m\}$  – это множество меток классов, где  $m$  – число классов. Как правило, метки классов понимаются как номера классов  $\Omega = \{1, 2, \dots, m\}$ . Классифицировать объект  $t \in T$  относительно классов с метками  $\omega_1, \omega_2, \dots, \omega_m$  – это значит найти т.н. индикаторную функцию  $g: T \rightarrow \Omega$ , которая ставит в соответствие объекту  $t \in T$  метку  $\omega_i$  того класса, которому он принадлежит, т.е.  $g(t) = \omega_i$ , если объект  $t$  принадлежит классу с меткой  $\omega_i$ .

Реально мы имеем дело не со всем множеством объектов  $T$ , а только с пространством признаков  $\Theta = P(T)$ . Тогда требуется найти такую функцию  $\tilde{g} : \Theta \rightarrow \Omega$ , которая ставила бы в соответствие каждому вектору  $y_i = P(t) \in \Theta$  метку  $\omega_i$  того класса, которому принадлежит соответствующий объект. Такая функция называется решающей. Часто решающую функцию называют решающим правилом.

В классической задаче распознавания образов решение о классе принимается только для отдельно взятого объекта. Хотя решающее правило многократно применяется к последовательности объектов, порядок их предъявления, либо другие взаимные связи между ними никак не учитываются. Фактически предполагается, что предъявляемый объект реально существует отдельно от других объектов. По этой причине априорное предположение о независимости объектов, образующих распознаваемое множество, оказывается очень естественным [7, 8].

## **1.2 Задача распознавания в массивах взаимосвязанных данных и минимизация среднего риска**

В классической задаче распознавания образов объекты, подлежащие распознаванию, рассматриваются независимо друг от друга.

В то же время, всегда существовали практические задачи [8], в которых множество всех объектов, подлежащих распознаванию, составляет единый массив, обусловленный природой исследуемого явления или процесса. В [8, 9] был рассмотрен один достаточно широкий класс задач анализа данных, решение которых связано с необходимостью анализировать массивы, образованные элементарными составляющими одинаковой природы, упорядоченными вдоль осей одного или нескольких аргументов. К объектам такого массива очевидным образом применимы понятия «смежности», «соседства», «упорядоченности». В частности, это приводит к тому, что порядок предъявления объектов становится чрезвычайно важным. Как следствие, возникает отсутствующее в классической задаче распознавания образов принципиально

новое свойство распознаваемого множества – взаимосвязанность составляющих его объектов.

В частности, сигналы естественно понимать как массивы данных, упорядоченные вдоль оси одного аргумента. Термин «сигнал» традиционно обозначает физическую величину, изменяющуюся во времени. С формальной точки зрения [15] сигнал есть функция скалярного аргумента  $y_t: T \rightarrow \Theta$ , принимающая значения в некотором множестве  $y_t \in \Theta$ , вообще говоря, произвольной природы. При компьютерном анализе неизбежно полагается, что аргумент  $t \in T$  пробегает дискретное множество значений в пределах действительной оси  $T = \{t_1, \dots, t_N\} \subset \mathbf{R}$ . Поскольку множество значений аргумента фиксировано, то, как правило, достаточно рассматривать порядковые номера упорядоченных значений аргумента  $t = 1, \dots, N$ . Тогда множество этих значений  $T = \{1, \dots, N\}$  естественно понимать как множество индексов элементов упорядоченного массива данных  $Y = (y_t, t = 1, \dots, N)$ . В задаче анализа предъявленного сигнала данных практически требуется для предъявленного сигнала  $Y = (y_t, t = 1, \dots, N)$ ,  $y_t \in \Theta$ , подобрать наиболее подходящую модель в виде некоторого другого сигнала  $X = (x_t, t = 1, \dots, N)$ , принимающего значения из некоторого другого множества  $x_t \in \Omega$ , которое определяется спецификой каждой прикладной задачи [15].

Выбор модели всегда производится из соображений компромисса между двумя требованиями, обычно противоречащими друг другу. С одной стороны, накопленное знание, по крайней мере, основных механизмов изучаемого явления, как правило, дает представление о том, какие модели более естественны и, следовательно, более ожидаемы, чем другие. Соответствующую информацию о предпочтении одних моделей перед другими, имеющуюся еще до того, как сигнал, подлежащий анализу, стал доступен наблюдателю, принято называть априорной информацией. С другой стороны, это же знание позволяет количественно судить, какие модели лучше согласуются с анализируемым сигналом по сравнению с другими моделями. Соответст-

вующую информацию, базирующуюся на анализе зарегистрированного сигнала, называют информацией наблюдения [15].

Часто термин «сигнал» используется в существенно более широком смысле, когда под сигналом понимается всякий массив числовых данных, упорядоченный вдоль оси некоторой переменной, не обязательно времени, например, это может быть пространственная координата или частота.

Упорядоченная совокупность сигналов образует уже двумерный массив. Распространенными двумерными массивами взаимосвязанных данных являются изображения на дискретном растре, необходимость анализа которых возникает в огромном числе приложений.

Трехмерные массивы образуются, например, из последовательности изображений, получаемых с некоторым шагом по времени. Возникают задачи, требующие анализа и четырехмерных массивов, например, когда один и тот же участок земной поверхности многократно фотографируется спутником с разных точек траектории и на разных витках.

Упорядоченность массивов данных вносит существенную специфику в методы их анализа. С одной стороны, заранее известное отношение порядка между элементами массива несет существенную априорную информацию о результате анализа, сокращая степень его неопределенности. Наличие естественной априорной информации упрощает анализ таких массивов в том смысле, что создает принципиальную возможность значительного повышения статистической стабильности результата. Но с другой стороны, эта же упорядоченность усложняет решение задачи с вычислительной точки зрения, поскольку появляется проблема разработки специальных алгоритмов для учета этой информации в процессе обработки массива.

В [15] была сформулирована задача распознавания образов в массивах данных, которая является непосредственным расширением задачи анализа предъявленного сигнала данных на двумерное поле носителя. В [7–9] была предложена обобщенная модель порождения объектов в поле многомерного носителя.

Пусть  $t \in T$  – элемент массива данных  $T$ . Пусть взаимосвязанность массива выражается в том, что на множестве его элементов задано антирефлексивное симметричное бинарное отношение  $G \subset T \times T$ , которое названо [7, 8] отношением соседства. Такое бинарное отношение определяет пары смежных элементов. Если элементы исходного массива  $t \in T$  и  $s \in T$  являются соседними, то имеет место  $(s, t) \in G$ . Данное отношение соседства представимо в виде неориентированного графа без петель, ребра которого соединяют соседние элементы массива данных (рис. 1.1). Важно отметить, что хотя обычное бинарное отношение соседства рефлексивно, в данном случае это не так, т.к.  $(t, t) \notin G$ .

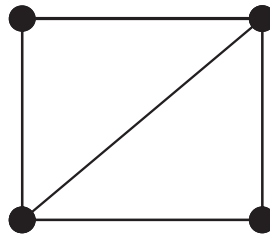


Рисунок 1.1 – Пример графа соседства массива, состоящего из 4 элементов

На рис. 1.2 приведены обычные графы соседства для сигнала, двумерного изображения и трехмерного куба данных.

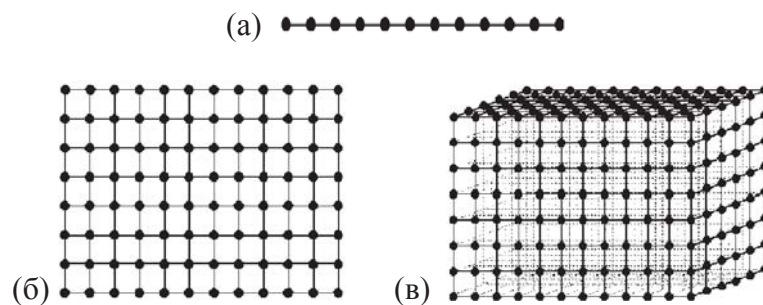


Рисунок 1.2 – Графы соседства элементов взаимосвязанных массивов: а) цепь; б) плоская решетка; в) трехмерная решетка (взято из [38])

В задаче распознавания взаимосвязанных объектов требуется по вектору наблюдаемых признаков  $y$ , определить принадлежность каждого элемента  $t \in T$  массива данных  $T$  к одному из классов  $\Omega$  с учетом априорной информации о преимущественных сочетаниях классов смежных объектов, заданных графом  $G$ .

В вероятностной постановке такой задачи распознавания предполагается, что массив данных представлен двухкомпонентным случайным полем, где наблюдаемая компонента  $Y$  состоит из векторов  $y_t$ , заданных на множестве элементов массива  $t \in T$  и принимающих значения из некоторого подходящего множества  $y_t \in \Theta$ , определенного природой источника данных. Элементы  $x_t, t \in T$  скрытой компоненты  $X$  принимают значения из некоторого множества  $x_t \in \Omega$ , специфичного для конкретной задачи анализа данных. Для задачи распознавания образов  $\Omega = \{1, 2, \dots, m\}$  – это конечное множество номеров классов объектов.

Соседство элементов двухкомпонентного случайного поля  $(X, Y)$  представлено неориентированным графом  $G$  без петель, ребра которого  $(s, t) \in G$  соединяют соседние элементы массива данных  $s \in T$  и  $t \in T$ .

Задача обработки массива  $(X, Y)$  представлена как задача преобразования исходного массива  $Y = (y_t, t \in T)$  в результирующий массив  $X = (x_t, t \in T)$ .

Такая задача была названа задачей распознавания массивов взаимосвязанных данных [7–9].

Пусть  $|T|$  – число элементов массива данных, тогда декартовы степени  $\Theta^{|T|}$  и  $\Omega^{|T|}$  образуют множества всех комбинаций значений исходных и результирующих (целевых) переменных, представляя множество всех массивов данных и множество результатов их обработки.

Пусть на множествах значений  $\Omega$  и  $\Theta$  определены некоторые  $\sigma$ -алгебры подмножеств и  $\sigma$ -конечные меры на них, позволяющие выразить вероятностные меры как на  $\Omega$  и  $\Theta$ , так и на  $\Omega^{|T|}$  и  $\Theta^{|T|}$  в виде соответствующих плотностей распределения вероятностей. Тогда априорное распределение вероятностей на множестве комбинаций целевых переменных имеет вид априорной плотности  $\zeta(X)$ ,  $X \in \Omega^{|T|}$ , а их вероятностная связь с исходными переменными выражена как условная плотность  $\eta(Y | X)$ ,  $Y \in \Theta^{|T|}$ .

Совместная плотность распределения скрытой и наблюдаемой компонент  $H(X, Y)$  определяется выражением

$$H(X, Y) = \zeta(X)\eta(Y | X), \text{ где } (X, Y) \in \Omega^{|\mathcal{T}|} \times \Theta^{|\mathcal{T}|}. \quad (1.1)$$

Задача обработки сводится к численному определению апостериорной плотности распределения вероятностей

$$\pi(X | Y) = \frac{H(X, Y)}{f(Y)} = \frac{\zeta(X)\eta(Y | X)}{f(Y)} \propto \zeta(X)\eta(Y | X). \quad (1.2)$$

Пусть  $\hat{X}(Y): \Theta^{|\mathcal{T}|} \rightarrow \Omega^{|\mathcal{T}|}$  – некоторый оператор, рассматриваемый как способ оценивания скрытого поля по зафиксированному наблюдаемому полю. Естественно, что оценка  $\hat{X}(Y)$ , вообще говоря, не будет совпадать с истинной реализацией скрытого поля  $X$ , появившейся совместно с наблюдаемым полем  $Y$  в составе двухкомпонентного поля  $(X, Y)$ .

В теории оценивания степень нежелательности такого несовпадения принято измерять посредством двухместной действительной функции  $\lambda(X, \hat{X}) \geq 0: \Omega^{|\mathcal{T}|} \times \Omega^{|\mathcal{T}|} \rightarrow \mathbf{R}$ ,  $\lambda(X, X) = 0$ , называемой функцией потерь. Предполагается, что функция потерь задана извне и отражает предпочтения пользователя конструируемого оператора оценивания.

Для всякого фиксированного оператора оценивания  $\hat{X}(Y)$  определены априорная плотность распределения вероятностей  $\zeta(X)$  и условная плотность  $\eta(Y | X)$ , определяющие совместную плотность распределения скрытой и наблюдаемой компонент  $H(X, Y)$  (1.1). Тогда полностью определена также случайная пара  $[X, \hat{X}(Y)]$ , и, следовательно, случайная величина потерь от несовпадения истинного значения скрытой компоненты и ее оценки  $\lambda[X, \hat{X}(Y)]$ . Математическое ожидание этой случайной величины  $\mathbf{M}\{\lambda[X, \hat{X}(Y)]\}$  называется средним риском ошибки.

Средний риск ошибки определяется выбором оператора оценивания  $\hat{X}(Y): \Theta^{|\mathcal{T}|} \rightarrow \Omega^{|\mathcal{T}|}$ , где аргумент  $Y$  пробегает все возможные комбинации

$Y \in \Theta^{[T]}$ . Поскольку с символом  $Y$  связывают конкретную комбинацию  $Y \in \Theta^{[T]}$ , то оператор в целом обозначается символом  $\hat{X}(\bullet)$  для всей совокупности наблюдений  $Y: \hat{X}(\bullet) = \{\hat{X}(Y), Y \in \Theta^{[T]}\}$ , где  $\hat{X}(Y) \in \Omega^{[T]}$ . Очевидно, что по своей математической структуре средний риск есть функционал на этом множестве:

$$r[\hat{X}(\bullet)] = M\{\lambda[X, \hat{X}(Y)]\} = \int_{Y \in \Theta^{[T]}} \int_{X \in \Omega^{[T]}} \lambda[X, \hat{X}(Y)] H(X, Y) dX dY. \quad (1.3)$$

Принцип минимизации среднего риска означает выбор в качестве оператора оценивания того из возможных операторов  $\hat{X}(Y)$ , который обеспечивает наименьшее значение условного среднего риска ошибки для наблюдения  $Y$ :

$$\hat{X}^*(Y) = \arg \min_{\hat{X}(Y) \in \Omega^{[T]}} r[\hat{X}(\bullet)] = \arg \min_{\hat{X}(Y)} \int_{X \in \Omega^{[T]}} \lambda[X, \hat{X}(Y)] \zeta(X) \eta(Y | X) dX. \quad (1.4)$$

Такой оператор оценивания скрытой переменной называют байесовским оператором, или, в сокращенной терминологии, принятой в математической статистике, байесовской оценкой.

Т.к.  $H(X, Y) = \zeta(X) \eta(Y | X) = p(Y) \pi(X | Y)$ , то для минимизации критерия (1.4) достаточно минимизировать следующий интеграл:

$$\hat{X}^*(Y) = \arg \min_{\hat{X} \in \Omega^{[T]}} \int_{X \in \Omega^{[T]}} \lambda[X, \hat{X}(Y)] \pi(X | Y) dX. \quad (1.5)$$

Таким образом, плотность апостериорного распределения вероятностей на множестве реализаций скрытого поля  $\pi(X | Y)$  полностью определяет байесовскую оценку  $\hat{X}^*(Y)$ .

Если полностью определены априорная плотность распределения вероятностей  $\zeta(X)$  и условная плотность  $\eta(Y | X)$ , а также функция потерь  $\lambda(X, \hat{X})$ , то задача построения алгоритма обработки данного вида сводится к нахождению байесовского оператора оценивания, т.е. к вычислению плотность апостериорного распределения (1.2) и решению задачи оптимизации (1.5).



Как правило, в задаче распознавания образов рассматриваются случаи сингулярной и аддитивной функций потерь.

Сингулярная функция потерь основана на понятии дельта-функции Дирака  $\delta(X, \hat{X})$ :

$$\lambda(X, \hat{X}) = 1 - \delta(X, \hat{X}), \text{ где } \delta(X, \hat{X}) = \begin{cases} 1, & \text{если } X = \hat{X}, \\ 0, & \text{если } X \neq \hat{X}. \end{cases}$$

Для сингулярной функции потерь интеграл в байесовском операторе оценивания (1.5) принимает вид:

$$\int_{X \in \Omega^T} \lambda(X, \hat{X}) \pi(X | Y) dX = \int_{X \in \Omega^T} \pi(X | Y) dX - \int_{X \in \Omega^T} \delta(X, \hat{X}) \pi(X | Y) dX.$$

Поскольку  $\int_{X \in \Omega^T} \pi(X | Y) dX = 1$ , то:

$$\int_{X \in \Omega^T} \lambda(X, \hat{X}) \pi(X | Y) dX = 1 - \pi(\hat{X}(Y) | Y). \quad (1.6)$$

Из (1.5) и (1.6) следует, что для сингулярной функции потерь оптимальной является оценка, максимизирующая апостериорную плотность распределения:

$$\hat{X}^*(Y) = \arg \max_{X \in \Omega^T} \pi(X | Y). \quad (1.7)$$

Сингулярная функция потерь измеряет несовпадение истинной реализации скрытого поля  $X = (x_t, t \in T)$  и ее оценки  $\hat{X} = (\hat{x}_t, t \in T)$  по всем элементам массива данных, не делая различий между ошибками в отдельных элементах массива. Однако в целом ряде прикладных задач анализа важно иметь возможность управлять желаемой точностью оценивания в разных элементах массива. Таковую возможность дает класс аддитивных функций потерь [15].

Пусть  $x_t \in \Omega$  – истинное и  $\hat{x}_t \in \Omega$  – оцененные значения скрытого процесса в отдельном элементе массива  $t \in T$ . Пусть в каждом элементе массива определена мгновенная функция потерь  $\mu_t(x, \hat{x}) \geq 0: \Omega \times \Omega \rightarrow \mathbf{R}$ ,  $\mu_t(x, x) = 0$ , измеряющая величину потерь от несовпадения значений  $\hat{x}_t$  и  $x_t$ . Общие по-

тери по всем элементам массива измеряются как сумма всех мгновенных потерь

$$\lambda(X, \hat{X}) = \sum_{t=1}^{|T|} \mu_t(x_t, \hat{x}_t). \quad (1.8)$$

Функция потерь  $\lambda(X, \hat{X})$  (1.8) называется аддитивной.

Для аддитивной функции потерь интеграл в байесовском операторе оценивания (1.5) принимает вид:

$$\int_{X \in \Omega^{|T|}} \lambda(X, \hat{X}) \pi(X | Y) dX = \sum_{t=1}^{|T|} \int_{X \in \Omega^{|T|}} \mu_t(x_t, \hat{x}_t) \pi(X | Y) dX. \quad (1.9)$$

В [15] было показано, что (1.9) имеет следующий эквивалентный вид:

$$\int_{X \in \Omega^{|T|}} \lambda(X, \hat{X}) \pi(X | Y) dX = \sum_{t=1}^{|T|} \int_{x_t \in \Omega} \mu_t(x_t, \hat{x}_t) p_t(x_t | Y) dx_t, \quad (1.10)$$

где  $p_t(x_t | Y)$  – апостериорная маргинальная плотность распределения вероятностей скрытых классов в элементе массива  $t \in T$ .

Предполагается, что мгновенная функция потерь является сингулярной

$$\mu_t(x_t, \hat{x}_t) = \begin{cases} 0, & x_t = \hat{x}_t, \\ 1, & x_t \neq \hat{x}_t. \end{cases}$$

Тогда (1.10) имеет следующий вид:

$$\int_{X \in \Omega^{|T|}} \lambda(X, \hat{X}) \pi(X | Y) dX = \sum_{t=1}^{|T|} (1 - p_t(\hat{x}_t | Y)) = |T| - \sum_{t=1}^{|T|} p_t(\hat{x}_t | Y). \quad (1.11)$$

Таким образом, согласно (1.11), байесовская оценка скрытого поля  $\hat{X}^*(Y)$ , представляющая собой массив оценок его значений  $\hat{X}^*(Y) = [\hat{x}_t^*(Y), t \in T]$ , определяется выражением

$$\hat{X}^*(Y) = [\hat{x}_t^*(Y), t \in T] = \arg \max_{\hat{x}_t \in \Omega} \sum_{t=1}^{|T|} p_t(\hat{x}_t | Y), t \in T. \quad (1.12)$$

Очевидно, что максимум достигается, если обеспечен максимум в каждом слагаемом в отдельности:

$$\hat{x}_t^*(Y) = \arg \max_{\hat{x}_t \in \Omega} p_t(\hat{x}_t | Y), t \in T. \quad (1.13)$$

В итоге, для аддитивной функции потерь байесовская оценка скрытой компоненты двухкомпонентного случайного поля распадается на совокупность отдельных оценок значений скрытой компоненты в каждом элементе массива данных.

Для удобства в дальнейшем под обычным обозначением оператора оценивания будем понимать байесовское решающее правило. Таким образом, для решения о скрытом поле в задаче распознавания образов в массивах взаимосвязанных данных часто используется байесовское решающее правило, которое может иметь вид

$$\hat{X}(Y) = \arg \max_{X \in \Omega^{|T|}} \pi(X | Y), \quad (1.14)$$

или

$$\hat{x}_t(Y) = \arg \max_{x_t \in \Omega} p_t(x_t | Y), t \in T. \quad (1.15)$$

В данной работе мы будем решать задачу распознавания образов в массивах взаимосвязанных на основе байесовского решающего правила вида (1.15).

### 1.3 Распознавание образов в массивах взаимосвязанных данных в случае ациклического графа соседства

В [7–9] вероятностная задача распознавания образов в массивах взаимосвязанных данных была сформулирована для ациклического графа соседства. В такой вероятностной постановке задачи для случая ациклического графа целью обработки по-прежнему является численная оценка апостериорных распределений (1.15).

Согласно [7–9], граф соседства  $G$  является ациклическим, т.е. не содержит циклов (рис. 1.3), а скрытое поле классов  $X$  является марковским относительно графа  $G$ .

Согласно [7–9], подмножества переменных при удалении из  $T$  и  $X$  одного элемента  $t$  и  $x_t$  обозначаются как  $T_{(t)} = (s \in T, s \neq t)$  и  $X_{(t)} = (x_s : s \in T_{(t)})$ , а подмножества переменных, смежных с элементом  $t$  в

графе  $G$ , соответственно как  $T_{(t)}^0 = (s \in T_{(t)}, (s, t) \in G)$  и  $X_{(t)}^0 = (x_s : s \in T_{(t)}, (s, t) \in G)$ . Аналогичные обозначения относительно графа  $G$  используются для соответствующих подмножеств элементов  $Y$ .

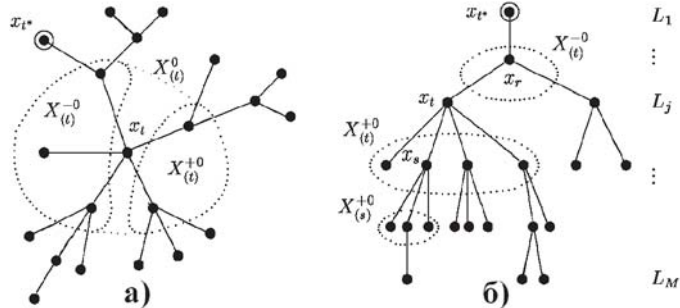


Рисунок 1.3 – Ациклический граф соседства (взято из [7])

Скрытое случайное поле  $X$  является марковским относительно графа  $G$ , если для любого  $t \in T$  условное распределение элемента  $x_t$  относительно остальных элементов  $q_t(x_t | X_{(t)})$  зависит только от значений смежных с ним элементов  $X_{(t)}^0$ :  $q_t(x_t | X_{(t)}) = q_t(x_t | X_{(t)}^0), t \in T$ .

Древовидный граф  $G$  разбивает окрестность (см. рис. 1.3а) нетерминального элемента  $x_t$  на две произвольные части  $X_{(t)}^0 = (X_{(t)}^{-0}, X_{(t)}^{+0})$ . Любая вершина  $t^* \in T$  в качестве корня немедленно задает естественный нисходящий порядок просмотра вершин от корня и восходящий порядок к корню, определяя окрестность  $X_{(t)}^{-0} = x_r$  из одного элемента, предшествующего  $x_t$ , и окрестность  $X_{(t)}^{+0}$  непосредственных потомков  $x_t$  (см. рис. 1.3б). В [7–9] показано, что такое априорное случайное поле является односторонним марковским  $q_t(x_t | X_{(t)}^0) = q_t(x_t | x_r)$ .

Согласно [7–9] также предполагается, что наблюдаемое поле  $Y = (y_t, t \in T)$  образовано случайными векторами  $y_t$ , условно независимыми относительно скрытого поля  $X = (x_t, t \in T)$ , где условные вероятностные свойства каждого из них полностью определяются только значением соответствующей скрытой переменной и выражаются условной плотностью распределения  $\psi_t(y_t | X) = \psi_t(y_t | x_t)$ . В [7–9] показано, что апостериорное скры-

тое случайное поле относительно того же графа  $G$  остается односторонним марковским  $p_t(x_t | X_{(t)}, Y) = p_t(x_t | x_r, Y_t^+)$ , где  $Y_t^+$  – поддереву с корнем в  $y_t$ , включая его.

В этих условиях обработка массива данных выполняется в два этапа: сначала из каждого элемента массива  $y_t$  извлекается информация о значении соответствующей скрытой переменной в виде ее апостериорного маргинального распределения  $p_t(x_t | y_t)$ ,  $t \in T$ , а затем такие несогласованные решения согласуются между собой с учетом совместного априорного распределения  $\zeta(X)$ .

Для задач распознавания образов в массивах взаимосвязанных объектов естественным способом поиска апостериорных маргинальных распределений  $p_t(x_t | y_t)$  является обучение по данным учителя, когда вместе со значениями наблюдаемых переменных  $y_t$  известны и значения их номеров классов  $x_t$ .

Базовый алгоритм распознавания в массивах взаимосвязанных данных для ациклического графа соседства состоит из следующих шагов [9]:

1. Фиксируется корень  $t^*$  как естественное начало обработки и задается априорное распределение  $q_{t^*}(x_{t^*})$ ,  $x_{t^*} \in \Omega$ .

2. Нисходящим просмотром от корня для всех  $t \in T$  вычисляются априорные распределения классов

$$q_s(x_s) = \sum_{x_t \in \Omega} q_s(x_s | x_t) q_t(x_t), \quad x_s \in \Omega, \quad s \in T_{(t)}^{+0}.$$

3. Восходящим просмотром от терминальных вершин к корню вычисляются фильтрационные апостериорные маргинальные распределения классов

$$p_t(x_t | Y_t^+) \propto p_t(x_t | Y_{(t)}^+) p_t(x_t | y_t), \quad x_t \in \Omega, \quad t \in T,$$

$$p_t(x_t | Y_{(t)}^+) \propto \prod_s \sum_{x_s \in \Omega} p_s(x_s | Y_s^+) \frac{q_s(x_s | x_t)}{q_s(x_s)}, \quad s \in T_{(t)}^{+0},$$

где апостериорные маргинальные распределения  $p_t(x_t | y_t)$  получены на этапе независимого обучения. В терминальных вершинах  $t$  каждого уровня, начиная с  $L_M$ , выполнено  $p_t(x_t | Y_t^+) = p_t(x_t | y_t)$  (см. рис. 1.3б).

4. На последнем восходящем шаге распределение в корне опирается на все наблюдения  $p_{t^*}(x_{t^*} | Y_{t^*}^+) = p_{t^*}(x_{t^*} | Y)$ ,  $x_{t^*} \in \Omega$ . Это апостериорное маргинальное распределение значений корневой переменной называется интерполяционным и позволяет принять решение о классе  $\hat{x}_{t^*}(Y) = \arg \max_{x_{t^*} \in \Omega} p_{t^*}(x_{t^*} | Y)$ .

5. Нисходящим просмотром от корня для остальных объектов  $t \in T$  вычисляются интерполяционные апостериорные маргинальные распределения

$$p_s(x_s | Y) \propto \sum_{x_t \in \Omega} p_s(x_s | x_t, Y) p_t(x_t | Y), \quad x_s \in \Omega, \quad s \in T_{(t)}^{+0},$$

$$p_s(x_s | x_t, Y) \propto p_s(x_s | Y_s^+) q_t(x_t) \frac{q_s(x_s | x_t)}{q_s(x_s)},$$

и принимаются решения о классах  $\hat{x}_s(Y) = \arg \max_{x_s \in \Omega} p_s(x_s | Y)$ .

#### 1.4 Частная модель скрытого марковского случайного поля принадлежностей объектов к классам

С целью упрощения в [7–9] был введен ряд предположений о модели марковского случайного поля  $X$ .

Марковская модель априорного скрытого случайного поля классов  $X$  определяет его одностороннее свойство в виде условных распределений вероятностей на множестве значений переменной  $x_t \in \Omega$  относительно любой переменной  $x_r \in X_{(t)}^0$  из марковской окрестности  $x_t$ . Для каждой пары вершин  $r, t \in G$ , соединенных ребром в дереве  $G$ , формально всегда определена пара условных распределений вероятностей  $q_t(x_t | x_r)$  и  $q_r(x_r | x_t)$ . Выбор некоторой вершины дерева  $G$  в качестве корневой  $t^* \in T$  задает два направления просмотра этого дерева, объявляя одно из этих распределений «нисходящим» распределением, а другое «восходящим» [9].

В теории марковских цепей с конечным числом состояний  $m$  рассматриваются свойства бесконечных последовательностей испытаний со стационарными переходными вероятностями  $q_{ij}(x_t = j | x_{t-1} = i)$ , где  $x_t, x_{t-1} \in \Omega = \{1, \dots, m\}$ , не зависящими от числа испытаний [6].

**Предположение 1.** Марковская цепь, представляющая одностороннюю марковскую модель случайного поля  $X$ , является однородной.

Тогда односторонняя марковская модель случайного поля  $X$  определяется как однородная конечная марковская цепь с неизменными условными распределениями в прямом (восходящем) и обратном (нисходящем) направлениях:  $q_r(x_r | x_t) = q(x_r | x_t)$ ,  $q_t(x_t | x_r) = \bar{q}(x_t | x_r)$ ;  $t, r \in T$ ;  $x_t, x_r \in \Omega$ .

Такая модель случайного поля  $X$  полностью определена матрицами прямых  $Q(m \times m)$  и обратных  $\bar{Q}(m \times m)$  условных вероятностей переходов марковской цепи.

**Предположение 2.** Матрица  $Q$  прямых условных вероятностей переходов имеет одинаковые диагональные элементы и одинаковые недиагональные элементы.

$$Q = \begin{bmatrix} q & \frac{1-q}{m-1} & \frac{1-q}{m-1} & \dots & \frac{1-q}{m-1} \\ \frac{1-q}{m-1} & q & \frac{1-q}{m-1} & \dots & \frac{1-q}{m-1} \\ & & \dots & & \\ \frac{1-q}{m-1} & \frac{1-q}{m-1} & \frac{1-q}{m-1} & \dots & q \end{bmatrix} \quad (1.16)$$

Такая марковская цепь является эргодической и имеет финальное распределение  $p(x)$  на множестве значений скрытых переменных  $x \in \Omega = \{1, 2, \dots, m\}$ , которое удовлетворяет уравнению Колмогорова [6]:  $p = pQ$ , где  $p = (p(1), p(2), \dots, p(m))$ .

Тогда  $p(Q - I_m) = \mathbf{0}$ , где  $I_m$  – единичная матрица размера  $m \times m$ :

$$p(Q - I_m) = [p(1) \ p(2) \ \dots \ p(m)] \begin{bmatrix} q-1 & \frac{1-q}{m-1} & \frac{1-q}{m-1} & \dots & \frac{1-q}{m-1} \\ \frac{1-q}{m-1} & q-1 & \frac{1-q}{m-1} & \dots & \frac{1-q}{m-1} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{1-q}{m-1} & \frac{1-q}{m-1} & \frac{1-q}{m-1} & \dots & q-1 \end{bmatrix} = \mathbf{0}.$$

Отсюда следует:

$$p(i)(q-1) + \frac{1-q}{m-1} \sum_{j=1, j \neq i}^m p(j) = 0, \quad i = 1, 2, \dots, m.$$

Так как  $\sum_{j=1}^m p(j) = 1$ , то:  $p(i)(q-1) + \frac{1-q}{m-1}(1 - p(i)) = 0, \quad i = 1, 2, \dots, m.$

$$\text{Отсюда } p(i) = \frac{1}{m}, \quad i = 1, 2, \dots, m. \quad (1.17)$$

Таким образом, финальное распределение  $p(x)$  на множестве значений скрытых переменных  $x \in \Omega$  является равномерным.

Для неразложимой марковской цепи элементы матриц  $Q$  и  $\bar{Q}$  связаны через финальное распределение вероятностей  $p(x)$  на множестве значений скрытых переменных  $x \in \Omega$ :

$$\bar{q}(x_t | x_r) = \frac{p(x_t)}{p(x_r)} q(x_r | x_t).$$

Так как финальное распределение  $p(x)$  равномерно, то  $p(x_t) = p(x_r)$ .

Из этого следует:

$$\bar{q}(x_t | x_r) = q(x_r | x_t).$$

Тогда матрица  $Q$  совпадает с матрицей  $\bar{Q}$ . В итоге, у нас остается только одна матрица условных вероятностей переходов, которая является симметричной и дважды стохастичной:



$$Q = \bar{Q} = \begin{bmatrix} q & \frac{1-q}{m-1} & \frac{1-q}{m-1} & \dots & \frac{1-q}{m-1} \\ \frac{1-q}{m-1} & q & \frac{1-q}{m-1} & \dots & \frac{1-q}{m-1} \\ & & \dots & & \\ \frac{1-q}{m-1} & \frac{1-q}{m-1} & \frac{1-q}{m-1} & \dots & q \end{bmatrix} \quad (1.18)$$

Очевидно, что в односторонней марковской модели начальное априорное маргинальное распределение вероятностей  $q_t(x_t)$  можно задать в любой вершине  $t \in T$ , тогда соответствующие маргинальные распределения оказываются известными во всех остальных вершинах дерева  $G$ , в том числе и в его корне. Выбор корня  $t$  естественно связать с началом обработки, а начальное априорное распределение сделать корневым  $q_{t^*}(x_{t^*}), x_{t^*} \in \Omega$ .

**Предположение 3.** В качестве начального априорного распределения классов в корне  $t^*$  задается равномерное финальное распределение.

$$\text{Это значит, что: } q_{t^*}(x_{t^*}) = p(x_{t^*}) = \frac{1}{m}.$$

Тогда маргинальные априорные распределения во всех остальных элементах также являются равномерными.

$$q_s(x_s) = \sum_{x_t \in \Omega} q_s(x_s | x_t) q_t(x_t) = \frac{1}{m} \sum_{x_t \in \Omega} q_s(x_s | x_t) = \frac{1}{m}, x_s \in \Omega, s \in T_{(t)}^{+0}, t \in T. \quad (1.19)$$

Поэтому второй шаг в базовом алгоритме распознавания можно пропустить.

Таким образом, мы получаем частную модель марковского поля, выраженную через модель марковской цепи, определенную единой матрицей условных вероятностей переходов (1.18), которая задается одним значением диагонального элемента  $q$ . Такая частная модель позволяет вполне адекватно описать марковские случайные поля, свойства которых обладают существенным разнообразием по сравнению со свойствами марковских цепей.

Но для такой модели значение диагонального элемента  $q$  задавалось ранее в [7–9] эвристически без подбора его оптимального значения. Поэтому

возникает задача подбора диагонального элемента матрицы условных вероятностей переходов для заданного ациклического графа.

### 1.5 Итерационный алгоритм повторения ациклического графа и комбинирование ациклических графов соседства

Как правило, в прикладных задачах графы соседства элементов массива данных содержат циклы. Например, в растровых изображениях естественное отношение соседства пикселей является решеткой, которая не является ациклическим графом. Очевидно, что древовидная редукция такого графа соседства существенно искажает взаимосвязи элементов в массиве данных. Выполнение оптимальной редукции в общем случае требует разработки специальных алгоритмов, сложность которых сопоставима со сложностью самого алгоритма распознавания [38]. Было предложено скомпенсировать редуцированное множество взаимосвязей элементов увеличением числа ациклических графов соседства [10–12, 27]. Исходный граф соседства заменяется набором ациклических графов соседства так, чтобы каждый граф покрывал все множество вершин, а каждое ребро исходного графа вошло хотя бы в один ациклический граф. Например, для сегментации растровых текстурных изображений удобно использовать графы на рис. 1.4.



Рисунок 1.4 – Набор ациклических графов соседства (взято из [12])

Итерационный алгоритм для заданного ациклического графа соседства [10] заключается в многократном применении базового алгоритма распознавания. Основу этого алгоритма составляет итерационный пересчет апостериорных маргинальных вероятностей элементов скрытого случайного поля  $X$ . После первой итерации мы перейдем от маргинальных распределений  $p_t(x_t | y_t)$ , полученных ранее на этапе независимого обучения, к апостериор-

ным маргинальным распределениям  $p_t(x_t | Y), t \in T$ . На следующей итерации вместо распределений  $p_t(x_t | y_t)$  используются только что полученные на предыдущей итерации апостериорные распределения  $p_t(x_t | Y), t \in T$ . Алгоритм повторяется до тех пор, пока распределения  $p_t(x_t | Y), t \in T$  не перестанут меняться или начнут циклически повторяться.

В алгоритме комбинирования [11, 12] было предложено заменить неизвестные нам апостериорные маргинальные распределения вероятностей скрытых классов относительно исходного графа соседства линейной комбинацией распределений относительно некоторого набора ациклических графов соседства. Алгоритм комбинирования ациклических графов заключается в следующем [11, 12].

1. Для графа  $G_k$  из данного набора применяется базовый алгоритм распознавания. Такая процедура для каждого ациклического графа из набора сформирует в отдельности распределения  $p_t^k(x_t | Y), t \in T, k = 1, \dots, K$ , где  $K$  – число ациклических графов, и соответствующие решения о классах  $\hat{x}_t^k(Y), t \in T$ .

2. Чтобы получить окончательные решения о классах  $\hat{x}_t(Y), t \in T$ , вычисляются апостериорные распределения  $p_t(x_t | Y)$  в каждом элементе  $t \in T$  как взвешенные суммы апостериорных распределений  $p_t^k(x_t | Y), k = 1, \dots, K$ , где

$$p_t(x_t | Y) = \sum_{k=1}^K w_k p_t^k(x_t | Y), \quad w_k \geq 0, \quad \sum_{k=1}^K w_k = 1. \quad (1.20)$$

3. Повторение комбинирования. Найденные взвешенные апостериорные распределения  $p_t(x_t | Y)$  берутся в качестве исходных вместо распределений  $p_t(x_t | y_t)$  для каждого ациклического графа соседства в отдельности и вновь повторяется комбинирование.

4. Процедура останавливается, когда результат перестанет изменяться.

Такая процедура названа итерационным алгоритмом распознавания с комбинированием графов соседства элементов взаимосвязанного массива.

Число  $w_k \geq 0$ , являющее коэффициентом важности графа  $G_k$ , называется весом графа  $G_k$ .

Очевидно, что каждый ациклический граф отражает лишь некоторое подмножество пространственных связей элементов исходного массива данных. Поэтому возникает задача подбора весов графов для конкретного типа распознаваемых изображений [12]. Задача подбора весов ациклических графов формулируется следующим образом.

Пусть заданы текстурное изображение и данные о точной сегментации, например, текстурное изображение на рис. 1.5.

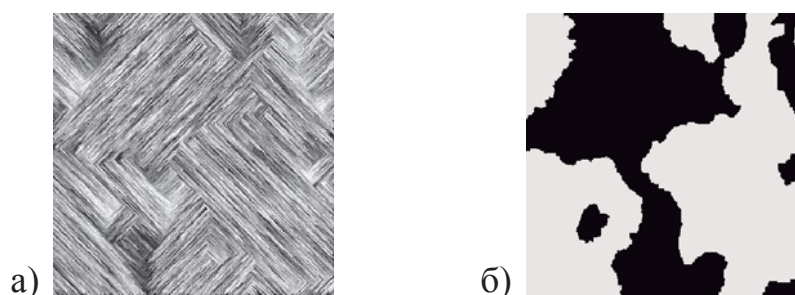


Рисунок 1.5 – Исходные данные: а) Текстурное изображение;  
б) Данные о точной сегментации

Текстурное изображение представляет собой массив данных  $T$ . Точки изображения – элементы массива данных  $t \in T$ . Для каждого элемента  $t \in T$  известен его номер класса  $x_t \in \Omega = \{1, 2, \dots, m\}$ .

Значение диагонального элемента матрицы условных вероятностей переходов фиксировано (например,  $q = 0.95$ ).

Пусть задано множество ациклических графов соседства  $G_1, G_2, \dots, G_K$  со своими весами  $w_1, w_2, \dots, w_K$  (например, множество графов на рис. 1.4), где

$$\sum_{k=1}^K w_k = 1, w_k \geq 0, k = 1, 2, \dots, K.$$

Число ошибок распознавания  $E(\mathbf{w})$ , зависящее от вектора весов графов  $\mathbf{w} = (w_1, w_2, \dots, w_K)$ , вычисляется как число элементов, для которых полученный и исходный номера классов не совпадают:

$$E(\mathbf{w}) = E(w_1, w_2, \dots, w_K) = |\Delta|, \text{ где } \Delta = \{t \in T, \hat{x}_t \neq x_t\} \text{ и}$$

$$\hat{x}_t = \hat{x}_t(Y) = \arg \max_{x_t \in \Omega} p_t(x_t | Y) = \arg \max_{x_t \in \Omega} \sum_{k=1}^K w_k p_t^k(x_t | Y). \quad (1.21)$$

Задача подбора весов ациклических графов заключается в поиске оптимального вектора весов  $\mathbf{w} = (w_1, w_2, \dots, w_K)$ , обеспечивающего минимальное число ошибок распознавания  $E(\mathbf{w})$ .

В [12] был предложен алгоритм определения весов при комбинировании ациклических графов соседства на основе известного алгоритма покоординатного спуска Гаусса-Зайделя. Алгоритм определения весов имеет следующий вид.

Аналогом покоординатного варьирования считается изменение веса графа  $G_k$  в диапазоне от 0 до 1.

На начальном шаге распределение весов всех графов изменяется от  $1/(K-1), \dots, 1/(K-1), 0, 1/(K-1), \dots, 1/(K-1)$ , когда граф  $G_k$  полностью исключен, до  $0, \dots, 0, 1, 0, \dots, 0$ , когда применяется только граф  $G_k$ . Шаг заканчивается после варьирования весов всех графов и выбора того графа, варьирование веса которого обеспечило минимальное число ошибок распознавания на некотором обучающем изображении.

На очередном шаге варьируется вес графа  $G_k$  в диапазоне  $0 \leq p \leq 1$ . Нормированный вес графа  $G_k$  в линейной комбинации имеет значение  $w_k = p$ . Остальные графы имеют к данному шагу постоянные веса  $l_i$ ,  $i = 1, \dots, K$ ,  $i \neq k$ , где их сумма также постоянна  $L = \sum_{i=1}^K l_i$ ,  $i \neq k$ . Их нормированные веса  $w_i = l_i(1-p)/L$  изменяются в диапазоне от  $w_i = l_i/L$  до  $w_i = 0$ .

Результат каждого пробного варьирования проверяется однократным комбинированием уже вычисленных маргинальных распределений  $p_t^k(x_t | Y)$ ,  $t \in T$ ,  $k = 1, \dots, K$  с подсчетом числа ошибок распознавания для решений  $\hat{x}_t$ ,  $t \in T$ . Шаг заканчивается после варьирования весов всех графов и выбора нового веса того графа, для которого получено минимальное число ошибок.

Следует отметить, что в алгоритме определения весов ациклических графов соседства при подсчете ошибок выполняется только один шаг итерационного алгоритма комбинирования ациклических графов, который также был назван однократным распознаванием или однократным комбинированием. Очевидно, что большое число шагов позволило бы в общем случае уменьшить число ошибок. Но это не было сделано, исходя из следующих соображений. С одной стороны, повышение числа шагов приводит к увеличению времени обучения. С другой стороны, не было желания слишком глубоко спускаться в локальный минимум. Целью являлась оценка статистического свойства решающего правила, а для этого не надо было слишком подстраиваться под конкретное изображение.

Также следует отметить, что в алгоритме определения весов ациклических графов была всего только одна матрица условных вероятностей переходов. Ее диагональный элемент так же задавался эвристически без поиска его оптимального значения.

Поэтому возникает новая задача одновременного подбора значения диагонального элемента и весов ациклических графов при их комбинировании. Далее в данной работе также предложено, чтобы каждому ациклическому графу соответствовала своя матрица условных вероятностей переходов, которая, в свою очередь, имеет свое значение диагонального элемента. Из этого также возникает еще одна задача подбора значений диагональных элементов и весов ациклических графов при их комбинировании.

Очевидно, что задача подбора весов ациклических графов соседства, дополненного подбором значений диагональных элементов, может иметь со-

вершено другую характеристику по сравнению с исходной задачей подбора весов графов. И этот факт может приводить к необходимости применения многократного распознавания (или многократного комбинирования) вместо однократного распознавания при подсчете числа ошибок на этапе обучения.

## **1.6 Натуральные и структурные параметры модели в задаче выбора модели**

Вообще говоря, параметры модели делятся на натуральные и структурные. Натуральные параметры модели отражают существенные свойства анализируемой системы. Натуральными параметрами являются, например, коэффициенты полинома в задаче аппроксимации кривой или параметры смеси гауссовских распределений в задаче кластеризации.

Тем не менее, как правило, в таких задачах, определяются и дополнительные параметры, которые либо используются для регуляризации конфигураций параметрической модели, либо представляют собой параметры априорных распределений, заданных для натуральных параметров. Такие параметры называют структурными (или гипер-параметрами). Другими словами, структурными называют параметры каких-то априорных предположений. В задаче выбора модели структурные параметры настраиваются, чтобы предотвратить переобучение. Примером структурного параметра [4, 19, 23] является коэффициент штрафа, дополненного до суммы отклонений в методе наименьших квадратов, или количество гауссианов в смеси нормальных распределений.

Задача определения корректных структурных параметров усложняется тем, что эти параметры по смыслу должны ограничивать допустимые решающие правила, чтобы избежать эффекта переобучения, и, следовательно, они не могут быть непосредственно включены в задачу минимизации ошибки на обучающей совокупности при оценке натуральных параметров. Существуют разные методы автоматической настройки структурных параметров. Среди них на сегодняшний день самым лучшим методом считается процедура скользящего контроля [4, 19, 21]. На практике такой метод может исполь-

зоваться для настройки лишь некоторых структурных параметров, но гипотетически он должен позволять определять любые характеристики наилучшей модели.

Принцип скользящего контроля состоит в построении критерия, который измеряет, насколько хорошо построенное решающее правило способно обобщить генеральную совокупность. Процедура скользящего контроля заключается в последовательном исключении части объектов из обучающей выборки, обучении на оставшихся объектах и распознавании исключенных объектов. В зависимости от способа исключения объектов возникают разные схемы скользящего контроля [4, 19, 21, 23].

Классической схемой скользящего контроля является схема *Leave-One-Out Cross Validation (LOOCV)*, где каждый объект последовательно исключается в качестве валидного множества, а остальные объекты используются для обучения. Построенное решающее правило проверяется на исключенном объекте и оценивается ошибка. Усредненное значение этих ошибок называется ошибкой скользящего контроля. *LOOCV* является весьма привлекательным и часто дает удовлетворительные результаты на практике. Однако такая схема часто оказывается вычислительно трудоемкой.

Более обобщенной схемой скользящего контроля является схема *Leave-p-out Cross Validation (LPOCV)*, в которой каждое возможное подмножество объектов размера  $p$  последовательно рассматривается как валидное множество, где  $1 \leq p \leq N-1$  и  $N$  – размер исходной совокупности. Если  $p=1$ , то схема *LPOCV* эквивалентна схеме *LOOCV*. Такой подход потенциально обладает гигантской вычислительной трудоемкостью, т.к. имеются всего  $C_N^p$  таких подмножеств размера  $p$ . Однако если  $p=N-1$ , то остается лишь  $N$  таких множеств. Кроме того, обучение выполняется только на одном объекте, что сильно снижает вычислительную трудоемкость. Таким образом, если алгоритм обучения является весьма вычислительно сложным, то схема *LPOCV*



с  $p = N - 1$  будет оказываться эффективной для оценки обобщенности решающего правила. Схему *LPOCV* с  $p = N - 1$  будем называть *неклассической*.

На практике часто применяется схема *k-folds Cross Validation*. Исходная совокупность  $N$  объектов разделяется на  $k$  равных групп по  $N/k$  объектов. Каждая группа последовательно исключается и используется в качестве валидного множества. Обучение выполняется на  $k - 1$  остальных группах. Построенное решающее правило проверяется на исключенной группе. Часто выбирают  $k = 5$  или  $k = 10$ .

При настройке структурных параметров по схеме скользящего контроля возможно переобучение на валидных данных. Ошибку на скользящем контроле нельзя рассматривать как оценку ошибки на генеральной совокупности, т.к. она получается заниженной. Поэтому часто используют другое множество, не участвующее в процедуре скользящего контроля. Это множество называется тестовым и на нем оценивается обобщенность выбранной модели [19, 21]. Размер тестового множества обычно составляет 20–30% объема генеральной совокупности.

Стандартным способом настройки структурных параметров является перебор по сетке, т.е. полный перебор всевозможных комбинаций значений структурных параметров и поиск набора значений, доставляющего минимум ошибке скользящего контроля.

Общая схема скользящего контроля для настройки структурных параметров выглядит следующим образом.

- разделить генеральную совокупность на два множества: обучающее (75%) и тестовое (25%);
- для каждой комбинации значений структурных параметров выполняется схема *k-folds Cross Validation* и подсчитывается ее ошибка. Оптимальные значения структурных параметров определяются в соответствии с минимальной ошибкой скользящего контроля;

- после нахождения оптимальных значений структурных параметров заново выполняется обучение на всех объектах обучающего множества. Решающее правило проверяется на тестовом множестве, вычисляя тестовую ошибку;

- как правило, разбиение генеральной совокупности повторяется много раз, например, 100 раз. Усредненная тестовая ошибка считается истинной ошибкой алгоритма на реальных данных. Значения параметров модели, как правило, также усредняются на основе результатов подбора по всем разбиениям.

Если  $k = N$ , то схема *k-folds Cross Validation* эквивалентна схеме *LOOCV*.

Проведение такой схемы требует значительного времени для многократного повторного обучения и применимо лишь для «быстрых» алгоритмов обучения. С помощью скользящего контроля можно настраивать не более двух-трех структурных параметров [4], т.к. настройка производится путем полного перебора всевозможных комбинаций параметров. Для уменьшения количества структурных параметров можно задать их в виде некоторого параметрического семейства. Однако, к сожалению, это не всегда удается сделать.

Но иногда различие между натуральными и структурными параметрами довольно нечетко. Отношение к параметру как структурному или натуральному может определяться точкой зрения исследователя. В частности, такая ситуация возникает в задаче подбора параметров комбинирования ациклических графов. Марковские параметры и веса ациклических графов могут рассматриваться как натуральные, так и структурные.

Таким образом, если считать веса ациклических графов и марковские параметры структурными, то задача настройки таких параметров является вычислительно трудоемкой. Нам удобнее будет считать веса графов натуральными параметрами модели.

Рассмотрим марковские параметры. Если рассматривать марковские параметры как натуральные, то их оценка как оценка натуральных параметров скрытой марковской модели [21, 42] может осуществляться по принципу максимального правдоподобия. В таком подходе параметры скрытой марковской модели оцениваются, как правило, EM-алгоритмом. Тем не менее, если оценивать статистические свойства решений рассмотренным способом [21, 42], то немедленно потребуются применить процедуру скользящего контроля и (что более принципиально) решить, какие из оцениваемых марковских параметров претендуют на роль структурных.

Если рассматривать марковские параметры как структурные, то необходимо решить проблему вычислительной трудоемкости из-за многократного повторения обучения при настройке структурных параметров по схеме скользящего контроля.

Чтобы избежать чрезмерной трудоемкости вычислительных экспериментов, в данной работе предложен «гибкий» подход, который заключается в подборе марковских параметров как натуральных, но качество решающего правила проверяется на основе схемы скользящего контроля, примененной при настройке структурных параметров. Для этого сначала мы построим алгоритмы подбора марковских параметров по отдельному изображению. Потом применим упрощенную схему скользящего контроля для выбора значений марковских параметров и для оценки ошибки на генеральной совокупности.

### **1.7 Минимизация гиббсовской энергии взаимодействия элементов марковского случайного поля**

Другим подходом к решению задачи распознавания образов в массивах взаимосвязанных данных является минимизация гиббсовской энергии.

В [31] введены понятия клики графа соседства и распределения Гиббса. Согласно [31], пусть  $T = \{t \in T\}$  – это множество элементов, и пусть  $G$  – граф соседства, который представляет собой отношения соседства между элементами множества  $T$ . Подмножество  $C \subseteq T$  называется кликой, если каждая

пара вершин в подмножестве  $C$  является соседними. Множество клик обозначается символом  $\mathcal{C}$ .

В частном случае, если в качестве графа соседства рассматривается решетка (см. рис. 1.2б) размера  $m \times m$ , то множество элементов  $T$  есть множество точек с координатами  $(i, j), 0 \leq i, j \leq m - 1$ .

Соседняя конфигурация относительно точки с координатами  $(i, j), 0 \leq i, j \leq m - 1$  при заданном ограничении  $c$  – это множество всех точек  $(k, l), 0 \leq i, j \leq m - 1$ , которые удовлетворяют условию:

$$0 < (k - i)^2 + (l - j)^2 \leq c. \quad (1.22)$$

Соседние конфигурации для случаев  $c = 1, 2, 8$  показаны на рис. 1.6.

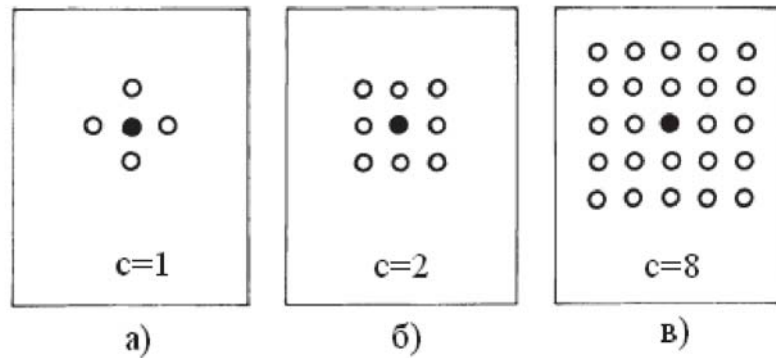


Рисунок 1.6 – Соседние конфигурации: а)  $c = 1$ ; б)  $c = 2$ ; в)  $c = 8$  (взято из [31])

Возможные клики в случаях  $c = 1, c = 2$  показаны на рис. 1.7. Очевидно, что количество возможных клик резко увеличивается с увеличением значения  $c$ .

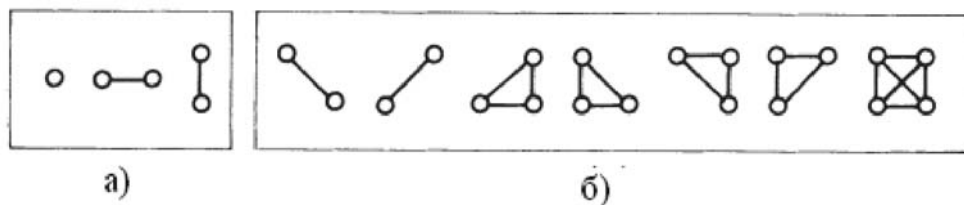


Рисунок 1.7 – Клики решетки: а)  $c = 1$ ; б)  $c = 2$  (взято из [31])

Если  $c = 1$ , то система соседства называется системой ближайших соседей, в которой существуют только одиночные клики и двоичные клики (рис. 1.7а).

Случайное поле  $X$  с некоторым графом соседства  $G$  произвольного вида называется гиббсовским случайным полем в том случае, если совместное распределение вероятностей его элементов является гиббсовским [31]. Гиббсовское распределение записывается в следующем виде:

$$\zeta(X) = \frac{1}{Z} e^{-\frac{1}{\tau} U(X)}, \quad (1.23)$$

где  $Z, \tau$  – константы, а  $U(X)$  называется функцией энергии гиббсовского распределения.

Величина  $Z$  называется также статистической суммой и служит для нормализации вероятностей  $\zeta(X), X \in \Omega^{|\Gamma|}$ :

$$Z = \sum_{X \in \Omega^{|\Gamma|}} e^{-\frac{1}{\tau} U(X)}. \quad (1.24)$$

Величина  $\tau$  называется температурой и контролирует остроту пика распределения.

Функция  $U(X)$  представляет собой сумму функций, определенных на кликах (полностью связанное подмножество вершин) графа соседства  $G$ :

$$U(X) = \sum_{C \in \mathcal{C}} V_C(X). \quad (1.25)$$

Функции  $V_C(X)$  называются также потенциалами клик графа. Значения потенциалов  $V_C(X)$  определяются значениями элементов на кликах  $C \in \mathcal{C}$ . Гиббсовское случайное поле является гомогенным, если  $V_C(X)$  не зависит от положения клики  $C \in \mathcal{C}$  в графе соседства  $G$ .

Решетка, представляющая собой естественное отношение соседства точек двумерного растрового изображений (см. рис. 1.2), соответствует системе ближайших соседней и содержит два типа клик, состоящих из одной и двух вершин. Соответствующие потенциалы могут быть обозначены следующим образом:  $V_1(x_t)$ , где  $x_t \in \Omega, t \in T$  и  $V_2(x_s, x_t)$ , где  $x_s, x_t \in \Omega; s, t \in T; (s, t) \in G$ . Энергия запишется в виде:

$$U(X) = \sum_{t \in T} V_1(x_t) + \sum_{t \in T} \sum_{s \in T_{(t)}^0} V_2(x_t, x_s); x_t, x_s \in \Omega. \quad (1.26)$$

Согласно известной теореме Хаммерсли-Клиффорда [37] об эквивалентности марковского и гиббсовского случайных полей, если случайное поле  $X$  является марковским относительно некоторого графа соседства  $G$ , то совместное распределение его элементов  $\zeta(X)$  является гиббсовским относительно того же самого графа соседства  $G$ .

Практическая ценность данной теоремы состоит в том, что она указывает простой способ указания совместного распределения для марковских случайных полей, заданных своими локальными характеристиками (условным распределением в элементе массива относительно соседних с ним элементов). Для задания совместного распределения  $\zeta(X)$  марковского случайного поля  $X$  необходимо лишь выбрать функции потенциалов на кликах, которые выражают априорные сведения о задаче.

Если совместное гиббсовское распределение поля данных задано (1.23), то мы можем вычислить условные распределения вероятностей в элементах массива относительно соседних элементов [31]:

$$p(x_t | X_{(t)}^0) = \frac{e^{-\sum_{C \in \mathcal{C}_t} V_C(X)}}{\sum_{x_t \in \Omega} e^{-\sum_{C \in \mathcal{C}_t} V_C(X)}}, \quad (1.27)$$

где  $C_t$  обозначает множество клик, содержащих вершину  $t$ .

Как было сказано, для определения гиббсовского распределения нам необходимо выбрать функции потенциалов клик. Этот выбор определяет поведение модели и должен производиться на основе априорных знаний о природе исследуемого явления.

Одной из известных моделей гиббсовских случайных полей является модель Поттса, которая является обобщением модели Изинга [34] и позволяет построить марковское случайное поле с  $m \geq 2$  состояниями, которые обычно обозначаются их номерами  $\Omega = \{1, 2, \dots, m\}$ .

В модели Поттса потенциалы клик задаются в следующем виде:

$$V_1(x_t) = \alpha_t x_t; \quad V_2(x_t, x_s) = \beta_{ts} I(x_t = x_s), \quad (1.28)$$

где  $I(i = j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$  – индикаторная функция, отмечающая совпадение значений ее аргументов.

В работе [31] было предложено восстанавливать скрытое случайное поле классов  $X$  на основе байесовского решающего правила для сингулярной функции потерь (1.14), а именно:  $\hat{X}(Y) = \arg \max_{X \in \Omega^T} \pi(X | Y)$ .

Согласно (1.2),  $\pi(X | Y) \propto \zeta(X)\eta(Y | X)$ , т.к. для конкретного массива наблюдений  $Y$  распределение  $f(Y)$  является константой.

Тогда то решающее правило (1.14) можно переписать в следующем виде:

$$\hat{X}(Y) = \arg \max_{X \in \Omega^T} [\zeta(X)\eta(Y | X)] \quad (1.29)$$

Если скрытое случайное поле  $X$  является марковским относительно графа соседства  $G$ , то, согласно теореме Хаммерсли-Клиффорда, распределение  $\zeta(X)$  тоже является гиббсовским относительно графа соседства  $G$ .

Согласно (1.23), мы имеем:  $\zeta(X) \propto e^{-U(X)}$  и  $\eta(Y | X) \propto e^{-U(Y|X)}$ .

Отсюда получаем, что  $\pi(X | Y) \propto e^{-(U(X)+U(Y|X))} = e^{-U(X|Y)}$ .

Решающее правило (1.29) запишется в виде:

$$\hat{X}(Y) = \arg \min_{X \in \Omega^T} U(X | Y). \quad (1.30)$$

Таким образом, в терминах гиббсовского распределения вероятностей задача восстановления скрытого случайного поля  $X$  по известной реализации наблюдаемого случайного поля  $Y$  сводится к отысканию минимума функции энергии апостериорного гиббсовского распределения.

Гиббсовский подход позволяет решать глобальную задачу минимизации энергии скрытых классов  $X$ . Он дает возможность рассматривать не только ациклические, но и циклические графы соседства (т.е. графы, которые содержат циклы). Теорема Хаммерсли-Клиффорда об эквивалентности гиббсовских и марковских случайных полей позволяет, используя апостериорные распределения, переводить их в термины гиббсовских энергий в окрест-

ностях. Однако у гиббсовского подхода есть и недостаток. В нем необходимо вводить эквивалент апостериорных распределений – гиббсовские энергии на кликах. Мы же по-прежнему хотим решать задачу непосредственно численной оценки апостериорных распределений. С этой целью в [7–9] для марковских полей было введено понятие односторонней окрестности, которое позволило использовать механизм марковских цепей. И поэтому, применяя комбинирование графов с подбором их весов, а также с подбором параметров марковских матриц условных вероятностей переходов, мы имеем возможность получить алгоритм такого же качества, как алгоритмы, построенные на поиске глобального экстремума, например, типа TRWS.

### **1.8 Основные цели и задачи исследования**

Предложенная в [7–9] древовидная модель марковского поля и алгоритмы распознавания оказались эффективными при обработке массивов взаимосвязанных данных. Но в такой модели диагональный элемент матрицы условных вероятностей переходов задавался эвристически без подбора его оптимального значения. Таким образом, задача поиска оптимального значения диагонального элемента оказывается актуальной для улучшения качества распознавания алгоритмами, построенными для древовидной модели марковского случайного поля.

Основными задачами исследования в данной работе являются:

1. Постановка задачи подбора диагонального элемента матрицы условных вероятностей переходов для ациклической модели и построение алгоритмов подбора диагонального элемента для заданного ациклического графа соседства.

2. Исследование свойств ранее предложенного в [12] алгоритма подбора весов при комбинировании ациклических графов соседства.

3. Постановка задачи одновременного подбора значения диагонального элемента матрицы условных вероятностей переходов и весов ациклических графов соседства при их комбинировании. Обобщение этой задачи для случая, когда каждому из ациклических графов соседства соответствует своя



марковская модель, которая, в свою очередь, определяется своим диагональным элементом матрицы условных вероятностей переходов. Разработка эффективных алгоритмов обучения распознаванию на основе подбора параметров комбинирования ациклических графов соседства.

4. Экспериментальное исследование разработанных алгоритмов в задаче сегментации растровых текстурных изображений. В качестве тестовых изображений используются как модельные, так и реальные изображения.

5. Оценка качества решения задачи распознавания в массивах взаимосвязанных данных и статистических свойств решающего правила методом скользящего контроля.

## 2 ЗАДАЧА ПОДБОРА ПАРАМЕТРОВ ДРЕВОВИДНОГО МАРКОВСКОГО СЛУЧАЙНОГО ПОЛЯ

### 2.1 Постановка задачи подбора диагонального элемента в случае ациклического графа соседства

В предыдущих работах были предложены алгоритм распознавания для ациклического графа соседства, алгоритм комбинирования графов и алгоритм определения весов графов. С целью упрощения этих алгоритмов была предложена частная модель скрытого марковского случайного поля в виде марковской цепи, которая задается своей матрицей условных вероятностей переходов, заданной одним значением ее диагонального элемента  $q$ . Значение  $q$  ранее задавалось эвристически. Целью данной работы является подбор оптимального значения диагонального элемента.

В задачах классификации стремятся к поиску решающего правила, при котором достигается наименьшая средняя вероятность ошибок (наименьший уровень ошибок). Ранее показано [7–9, 38, 39], что в задачах обработки взаимосвязанных массивов данных задача минимизации уровня ошибок сводится к задаче максимизации совместной апостериорной вероятности  $\pi(X|Y)$  в целом, или апостериорных вероятностей  $p_t(x_t|Y), t \in T$  для отдельных элементов массива, что приводит к байесовскому решающему правилу, которое описывается в следующем виде:

$$\hat{X}(Y) = \arg \max_{X \in \Omega^T} \pi(X|Y) \quad \text{или} \quad (2.1)$$

$$\hat{x}_t(Y) = \arg \max_{x_t \in \Omega} p_t(x_t|Y). \quad (2.2)$$

Пусть, как и ранее, наблюдаемое поле  $Y = (\mathbf{y}_t, t \in T)$  образовано случайными векторами  $\mathbf{y}_t$ , условно независимыми  $\psi_t(\mathbf{y}_t|X) = \psi_t(\mathbf{y}_t|x_t)$  относительно скрытого поля  $X = (x_t, t \in T)$ . Тогда апостериорное скрытое поле классов формально определяется по следующей формуле:

$$\pi(X|Y) \propto \frac{\zeta(X)}{\prod_{t \in T} q_t(x_t)} \prod_{t \in T} p_t(x_t|\mathbf{y}_t). \quad (2.3)$$

Для каждой реализации случайного поля  $(X, Y)$ , маргинальные апостериорные вероятности  $p_v(x_v | y_v), v \in T$  известны как результат независимого обучения. С другой стороны, в частной модели случайного марковского поля все маргинальные априорные распределения равномерны  $q_v(x_v) = 1/m, v \in T, x_v \in \Omega$ .

Таким образом, каждой реализации случайного поля  $(X, Y)$  задача максимизации апостериорной совместной вероятности  $\pi(X | Y)$  сводится к задаче максимизации априорного совместного распределения  $\zeta(X)$ .

Если скрытое случайное поле  $X$  является марковским, а граф соседства является ациклическим, то ранее показано [7–9], что для такого марковского поля всегда существует одностороннее представление, являющееся обобщенной марковской цепью:

$$\zeta(X) = \zeta(X_1, \dots, X_M) = P_1(X_1) \prod_{j=2}^M P_j(X_j | X_{j-1}). \quad (2.4)$$

Там же показано [7–9], что совместное распределение скрытых классов  $\zeta(X)$  можно вычислить по следующей формуле (см. рис. 1.3б):

$$\zeta(X) = q_{t^*}(x_{t^*}) \prod_{j=1}^{M-1} \prod_{t \in L_j} \prod_{s \in T_{(t)}^{+0}} q_s(x_s | x_t) = q_{t^*}(x_{t^*}) \prod_{s \in T_{(t^*)}^{-0}} q_s(x_s | x_t), t \in T_{(s)}^{-0}. \quad (2.5)$$

В частной модели случайного марковского поля условные вероятности переходов задаются в виде одной матрицы  $Q(m \times m)$  с одним значением диагонального элемента  $0 < q < 1$ :

$$Q = \begin{bmatrix} q & \frac{1-q}{m-1} & \frac{1-q}{m-1} & \dots & \frac{1-q}{m-1} \\ \frac{1-q}{m-1} & q & \frac{1-q}{m-1} & \dots & \frac{1-q}{m-1} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{1-q}{m-1} & \frac{1-q}{m-1} & \frac{1-q}{m-1} & \dots & q \end{bmatrix}, \text{ где } m - \text{число классов.}$$

Таким образом, для каждой реализации случайного поля  $X = (x_t, t \in T)$ , соответствующая априорная совместная вероятность  $\zeta(X)$  полностью определяется значением диагонального элемента  $q$  по формуле (2.5), где

$$q_s(x_s | x_t) = \begin{cases} q, & x_s = x_t, \\ \frac{1-q}{m-1}, & x_s \neq x_t. \end{cases} \quad (2.6)$$

Пусть  $V_1 = \{u \in T_{(t^*)} | x_u = x_v, v \in T_{(u)}^{-0}\}$ ,  $V_2 = \{u \in T_{(t^*)} | x_u \neq x_v, v \in T_{(u)}^{-0}\}$ .

Очевидно, что  $V_1 \cap V_2 = \emptyset$ ,  $V_1 \cup V_2 = T_{(t^*)}$ .

Тогда (2.5) имеет вид:

$$\zeta(X, q) = q_{t^*}(x_{t^*}) \prod_{s \in V_1} q_s(x_s | x_t) \prod_{s \in V_2} q_s(x_s | x_t) = q_{t^*}(x_{t^*}) q^{|V_1|} \left( \frac{1-q}{m-1} \right)^{|V_2|}, \quad t \in T_{(s)}^{-0}. \quad (2.7)$$

Таким образом, для данной реализации поля  $X = (x_t, t \in T)$  максимизация апостериорной совместной вероятности  $\pi(X | Y)$  сводится к максимизации априорной совместной вероятности  $\zeta(X)$ , что, в свою очередь, ведет к поиску значения диагонального элемента  $q$ , доставляющего максимум совместной априорной вероятности:

$$\hat{q}(X) = \arg \max_{0 < q < 1} \zeta(X, q). \quad (2.8)$$

Заметим, что формула (2.8), на самом деле, является принципом максимизации правдоподобия данной реализации  $X = (x_t, t \in T)$  относительно параметра  $q$ . Оценка по (2.8) дает диагонального элемента  $q$ , которое наиболее хорошо интерпретирует данную реализацию скрытого поля  $X = \{x_t, t \in T\}$ .

Для поиска экстремума (2.7) продифференцируем функцию  $\zeta(X, q)$  по

$$q: \quad \frac{d\zeta(X, q)}{dq} = \frac{d}{dq} q_{t^*}(x_{t^*}) q^{|V_1|} \left( \frac{1-q}{m-1} \right)^{|V_2|} = 0.$$

Тогда получим:

$$\left( \frac{1-q}{m-1} \right)^{|V_2|} |V_1| q^{|V_1|-1} + q^{|V_1|} |V_2| \left( \frac{1-q}{m-1} \right)^{|V_2|-1} \frac{-1}{m-1} = 0.$$

Следовательно:

$$\frac{1}{q} |V_1| - \frac{1}{1-q} |V_2| = 0.$$

В итоге:

$$q = \frac{|V_1|}{|V_1| + |V_2|} = \frac{|V_1|}{|T| - 1}. \quad (2.9)$$

Как правило, при выборе данных для обучения учитель указывает какие-нибудь области классов. Но тогда не хватает информации, чтобы непосредственно дать оценку значения диагонального элемента  $q$  по (2.9). Более того, разные варианты данных учителя дадут разные результаты обучения. Поэтому чтобы результат обучения не изменился, мы предположим, что учитель указал все точки классов изображения, т.е. задал точную сегментацию.

Если задана точная сегментация изображения, то на ее основе можно непосредственно оценить значение диагонального элемента  $q$  по (2.9). Такая оценка дает истинное значение диагонального элемента, которое наиболее хорошо интерпретирует заданную точную сегментацию изображения. Эксперименты показывают, что такая оценка  $q$  очень близко к единице. Более того, первое применение предложенных в [9, 11, 12] алгоритмов распознавания с таким значением диагонального элемента сильно уменьшает число ошибок распознавания, быстро сливая маленькие ошибочные области, полученные на этапе независимого обучения. Тем не менее, из-за такого быстрого слияния областей дальнейшее применение таких алгоритмов распознавания часто оказывается неэффективным.

Таким образом, непосредственная оценка диагонального элемента  $q$  дает его истинное значение, но не обеспечивает наилучшего качества работы предложенных в [9, 11, 12] алгоритмов распознавания.

Это противоречие объясняется тем, что предложенные в [9, 11, 12] алгоритмы были построены с целью корректировки результата независимого распознавания. На этапе независимого распознавания используется информация о наблюдениях и никакой информации об истинном значении марков-

ского параметра не учитывается. Поэтому, метод поиска марковского параметра, основанный на точной реализации скрытого поля, не совсем годится для предложенных в [9, 11, 12] алгоритмов распознавания и, в частности, и для базового алгоритма распознавания по ациклическому графу соседства [9].

Поэтому вместо точной сегментации, мы будем оценивать значение по промежуточной сегментации, полученной при решении задачи (2.2). Такой подход позволяет нам оценить значение диагонального элемента  $q$  даже в случаях, когда учитель частично указывает области классов для обучения.

Решая задачу (2.2) при фиксированных  $q$  и  $Y$ , получаем промежуточную сегментацию. На основе этой сегментации оценим  $q$  по (2.9). Можно предположить, что такая оценка диагонального элемента  $q$  получается заниженной по сравнению с его фактическим значением. Это позволяет эффективно повторять базовый алгоритм, избегая быстрого слияния областей классов.

В результате решения задачи (2.2) будет получена оценка скрытой компоненты  $\check{X}(Y) = \{\hat{x}_t(Y), t \in T\}$ . Тогда соответствующая оценка  $\hat{q}(\check{X})$  имеет следующий вид:

$$\hat{q}(\check{X}) = \arg \max_{0 < q < 1} \zeta(\check{X}, q). \quad (2.10)$$

Очевидно, что решение задачи (2.10) заключается в решении двух взаимосвязанных задач:

- а) при фиксированных  $q$  и  $Y$  решить задачу (2.2),
- б) при фиксированном  $X$  решить задачу (2.8).

Описанная схема решения заключается в итеративном решении задач (а) и (б).

## 2.2 Предельные значения диагонального элемента

Значение диагонального элемента  $q$  означает вероятность, с которой марковская цепь останется в текущем состоянии на следующем шаге. Поэтому вообще  $q$  может принимать значения в диапазоне от 0 до 1:

$$0 < q < 1. \quad (2.11)$$

Часто естественно предположить, что соседние элементы взаимосвязанного массива чаще принадлежат к одному классу, чем к разным, образуя достаточно протяженные области с относительно постоянными вероятностными свойствами наблюдаемого поля  $Y$  при достаточно простой форме однородных областей скрытого поля  $X$ , соответствующих разным классам. Это предположение реализуется выбором таких значений в марковской матрице условных вероятностей переходов, при которых порождаются последовательности испытаний, подряд принадлежащих одному состоянию, длительности которых соизмеримы с размером однородных областей в массиве данных. Поэтому в общем случае следует обеспечить только условия, при которых марковская цепь чаще будет оставаться в одном состоянии, чем переходить в другое, т.е. нужно рассмотреть только значения диагонального элемента  $q$ , удовлетворяющее условию:

$$q \geq \frac{1-q}{m-1}. \quad (2.12)$$

Из (2.11) и (2.12), получим:

$$\begin{cases} 0 < q < 1 \\ q \geq \frac{1-q}{m-1} \end{cases} \Rightarrow \begin{cases} 0 < q < 1 \\ q \geq \frac{1}{m} \end{cases} \Rightarrow \frac{1}{m} \leq q < 1. \quad (2.13)$$

Рассмотрим поведение базового алгоритма распознавания при предельных значениях диагонального элемента.

**Свойство 1.** Если значение диагонального элемента  $q = 1/m$ , то результат базового алгоритма полностью совпадает с независимым распознаванием.

Докажем, что в данном случае применение базового алгоритма не изменяет вероятности скрытых состояний, полученных на этапе независимого распознавания.

Пусть  $q = 1/m$ . Тогда  $q_s(x_s | x_t) = 1/m, x_s \in \Omega, x_t \in \Omega$ .

- Восходящий просмотр:

$$p_t(x_t | Y_t^+) \propto p_t(x_t | Y_{(t)}^+) p_t(x_t | \mathbf{y}_t), \quad x_t \in \Omega, \quad t \in T, \quad \text{где}$$

$$p_t(x_t | Y_{(t)}^+) \propto \prod_s \sum_{x_s \in \Omega} p_s(x_s | Y_s^+) \frac{q_s(x_s | x_t)}{q_s(x_s)}, \quad s \in T_{(t)}^{+0}.$$

$$\text{Тогда } p_t(x_t | Y_{(t)}^+) \propto \prod_s \sum_{x_s \in \Omega} p_s(x_s | Y_s^+) \frac{1/m}{1/m} = \prod_s \sum_{x_s \in \Omega} p_s(x_s | Y_s^+) = 1, \quad s \in T_{(t)}^{+0}.$$

$$\text{Откуда: } p_t(x_t | Y_t^+) \propto p_t(x_t | Y_{(t)}^+) p_t(x_t | \mathbf{y}_t) = p_t(x_t | \mathbf{y}_t).$$

- Нисходящий просмотр:

$$p_s(x_s | Y) \propto \sum_{x_t \in \Omega} p_s(x_s | x_t, Y) p_t(x_t | Y), \quad x_s \in \Omega, \quad s \in T_{(t)}^{+0}, \quad \text{где}$$

$$p_s(x_s | x_t, Y) \propto p_s(x_s | Y_s^+) q_t(x_t) \frac{q_s(x_s | x_t)}{q_s(x_s)} = p_s(x_s | Y_s^+) \frac{1}{m} \frac{1/m}{1/m} = \frac{1}{m} p_s(x_s | Y_s^+).$$

Однако,  $p_s(x_s | Y_s^+) = p_s(x_s | \mathbf{y}_s)$  (доказано при восходящем просмотре).

$$\text{Откуда: } p_s(x_s | x_t, Y) \propto \frac{1}{m} p_s(x_s | \mathbf{y}_s),$$

$$p_s(x_s | Y) \propto \sum_{x_t \in \Omega} p_s(x_s | x_t, Y) p_t(x_t | Y) = \sum_{x_t \in \Omega} \frac{1}{m} p_s(x_s | \mathbf{y}_s) p_t(x_t | Y) =$$

$$= \frac{1}{m} p_s(x_s | \mathbf{y}_s) \sum_{x_t \in \Omega} p_t(x_t | Y) = \frac{1}{m} p_s(x_s | \mathbf{y}_s), \quad \text{т.е. } p_s(x_s | Y) \propto p_s(x_s | \mathbf{y}_s).$$

Свойство 1 доказано.

**Свойство 2.** Если значение диагонального элемента  $q = 1$ , то фильтрационное распределение на множестве классов для каждого элемента массива эквивалентно произведению маргинальных апостериорных распределений, восстановленных на этапе обучения для всех элементов массива, принадлежащих поддереву с корнем в данном элементе:



$$p_t(x_t = i | Y_t^+) \propto \prod_{s \in T_t^+} p_s(x_s = i | \mathbf{y}_s), \text{ где } i \in \Omega = \{1, 2, \dots, m\}. \quad (2.14)$$

Докажем данное свойство по методу математической индукции.

Рассмотрим уровни дерева снизу вверх, т.е. по порядку  $L_M, L_{M-1}, \dots, L_1$  (см. рис. 1.3б).

- Очевидно, что все вершины уровня  $L_M$  являются терминальными.

Поэтому для вершины  $t \in L_M$ :

$$p_t(x_t = i | Y_t^+) = p_t(x_t = i | \mathbf{y}_t) = \prod_{s \in T_t^+} p_s(x_s = i | \mathbf{y}_s), i \in \Omega \text{ (свойство 2 верно).}$$

- Допустим, что свойство 2 верно до уровня дерева  $L_k$ ,  $k = 2, 3, \dots, M$ .

Пусть  $t$  – одна вершина на уровне дерева  $L_{k-1}$ .

Если вершина  $t$  является терминальной, то для ей свойство 2 верно (как показано выше).

Если вершина  $t$  не является терминальной, то:

$$\begin{aligned} p_t(x_t | Y_t^+) &\propto p_t(x_t | Y_{(t)}^+) p_t(x_t | \mathbf{y}_t) = p_t(x_t | \mathbf{y}_t) \prod_{s \in T_{(t)}^{+0}} \sum_{x_s \in \Omega} p_s(x_s | Y_s^+) \frac{q_s(x_s | x_t)}{q_s(x_s)} = \\ &= p_t(x_t | \mathbf{y}_t) \prod_{s \in T_{(t)}^{+0}} \sum_{x_s \in \Omega} p_s(x_s | Y_s^+) \frac{q_s(x_s | x_t)}{1/m} \propto p_t(x_t | \mathbf{y}_t) \prod_{s \in T_{(t)}^{+0}} \sum_{x_s \in \Omega} p_s(x_s | Y_s^+) q_s(x_s | x_t). \end{aligned}$$

Отсюда следует:

$$p_t(x_t = i | Y_t^+) \propto p_t(x_t = i | \mathbf{y}_t) \prod_{s \in T_{(t)}^{+0}} \sum_{x_s \in \Omega} p_s(x_s | Y_s^+) q_s(x_s | x_t = i), i \in \Omega. \quad (2.15)$$

Т.к.  $q = 1$ , то

$$q_s(x_s | x_t) = \begin{cases} 1, & x_s = x_t, \\ 0, & x_s \neq x_t. \end{cases} \quad (2.16)$$

Из (2.15) и (2.16), получим:

$$p_t(x_t = i | Y_t^+) \propto p_t(x_t = i | \mathbf{y}_t) \prod_{s \in T_{(t)}^{+0}} p_s(x_s = i | Y_s^+), i \in \Omega. \quad (2.17)$$

Т.к.  $t \in L_{k-1}, s \in T_{(t)}^{+0}; k = 2, 3, \dots, M$ , то  $s \in L_k$ . Поэтому:

$$p_s(x_s = i | Y_s^+) \propto \prod_{u \in T_s^+} p_u(x_u = i | \mathbf{y}_u), i \in \Omega \text{ (по индукции).}$$

Следовательно:

$$\begin{aligned}
p_t(x_t = i | Y_t^+) &\propto p_t(x_t = i | \mathbf{y}_t) \prod_{s \in T_t^{+0}} p_s(x_s = i | Y_s^+) \propto \\
&\propto p_t(x_t = i | \mathbf{y}_t) \prod_{s \in T_t^{+0}} \prod_{u \in T_s^+} p_u(x_u = i | \mathbf{y}_u) = \prod_{s \in T_t^+} p_s(x_s = i | \mathbf{y}_s).
\end{aligned}$$

Это означает, что свойство 2 также верно на уровне дерева  $L_{k-1}$ .

По принципу индукции свойство верно на всех уровнях дерева, т.е. для всех его вершин. Свойство 2 доказано.

**Следствие.** Если значение диагонального элемента  $q = 1$ , то интерполяционное апостериорное распределение на множестве классов корневого элемента массива эквивалентно произведению маргинальных апостериорных распределений на всех элементах массива:

$$p_{t^*}(x_{t^*} = i | Y) = p_{t^*}(x_{t^*} = i | Y_{t^*}^+) \propto \prod_{s \in T} p_s(x_s = i | \mathbf{y}_s), i \in \Omega. \quad (2.18)$$

**Свойство 3.** Если значение диагонального элемента  $q = 1$ , то интерполяционное апостериорное распределение на множестве классов каждого элемента массива вычисляется по следующей формуле:

$$p_s(x_s = i | Y) \propto \prod_{t \in T_s^-} \prod_{u \in T_t^+} p_u(x_u = i | \mathbf{y}_u), s \in T, i \in \Omega, \quad (2.19)$$

где  $T_s^-$  – множество предков элементов  $s \in T$ , включая его самого, относительно корня  $t^*$ .

Докажем данное свойство по методу математической индукции. Рассмотрим уровни дерева сверху вниз, т.е. по порядку  $L_1, L_2, \dots, L_M$  (см. рис.1.3б).

- Уровень  $L_1$  содержит только одну корневую вершину  $t^*$ . Для этой вершины на основании следствия свойства 2 справедливо:

$$p_{t^*}(x_{t^*} = i | Y) \propto \prod_{u \in T} p_u(x_u = i | \mathbf{y}_u) \propto \prod_{u \in T_{t^*}^+} p_u(x_u = i | \mathbf{y}_u) \propto \prod_{t \in T_{t^*}^- = \{t^*\}} \prod_{u \in T_t^+} p_u(x_u = i | \mathbf{y}_u)$$

(свойство 3 верно).

- Допустим, что свойство верно до уровня дерева  $L_k, k = 1, 2, \dots, M - 1$ .

Пусть  $s$  – одна вершина на уровне дерева  $L_{k+1}$ . А вершина  $t$  – его непосредственный предок, который принадлежит уровню дерева  $L_k$ .

Тогда:

$$p_s(x_s | Y) \propto \sum_{x_t \in \Omega} p_s(x_s | x_t, Y) p_t(x_t | Y) \propto \sum_{x_t \in \Omega} p_t(x_t | Y) p_s(x_s | Y_s^+) \frac{q_t(x_t)}{q_s(x_s)} q_s(x_s | x_t) \\ \propto \sum_{x_t \in \Omega} p_t(x_t | Y) p_s(x_s | Y_s^+) q_s(x_s | x_t) .$$

Так как

$$q_s(x_s | x_t) = \begin{cases} 1, & x_s = x_t, \\ 0, & x_s \neq x_t, \end{cases}$$

то

$$p_s(x_s = i | Y) \propto \sum_{x_t \in \Omega} p_t(x_t | Y) p_s(x_s = i | Y_s^+) q_s(x_s = i | x_t) \propto \\ \propto p_t(x_t = i | Y) p_s(x_s = i | Y_s^+), i \in \Omega. \quad (2.20)$$

Очевидно, что:

$$p_t(x_t = i | Y) \propto \prod_{v \in T_t^-} \prod_{u \in T_v^+} p_u(x_u = i | \mathbf{y}_u), \text{ т.к. } t \in L_k \text{ (по индукции)}. \quad (2.21)$$

$$p_s(x_s = i | Y_s^+) \propto \prod_{k \in T_s^+} p_k(x_k = i | \mathbf{y}_k), \text{ согласно свойству 1.} \quad (2.22)$$

Из (2.20), (2.21) и (2.22), получим:

$$p_s(x_s = i | Y) \propto \left( \prod_{v \in T_t^-} \prod_{u \in T_v^+} p_u(x_u = i | \mathbf{y}_u) \right) \left( \prod_{k \in T_s^+} p_k(x_k = i | \mathbf{y}_k) \right) \propto \\ \propto \prod_{v \in T_s^-} \prod_{u \in T_v^+} p_u(x_u = i | \mathbf{y}_u) .$$

Это означает, что свойство 3 также верно на уровне дерева  $L_{k+1}$ .

По принципу индукции свойство верно на всех уровнях дерева, т.е. для всех его вершин. Свойство 3 доказано.

Для однородной марковской цепи, если значение диагонального элемента  $q$  матрицы условных вероятностей переходов равно 1, то цепь всегда остается в одном состоянии. В задаче распознавания образов с множеством классов  $\Omega = \{1, 2, \dots, m\}$  такая марковская цепь дает большое число ошибок, математическое ожидание которых равно  $(m-1)/m$ .

Рассмотрим, например, цепь, как частный случай ациклического графа соседства. Пусть последовательность состоит из  $N$  вершин  $t_1, t_2, \dots, t_N$  (рис. 2.1).



Рисунок 2.1 – Последовательность вершин длины  $N$

Для этой последовательности априорное совместное распределение скрытых классов  $\zeta(X)$  определяется

$$\zeta(X) = \zeta(x_{t_1}, x_{t_2}, \dots, x_{t_N}) = p_{t_1}(x_{t_1}) \prod_{i=2}^N q_{t_i}(x_{t_i} | x_{t_{i-1}}). \quad (2.23)$$

Множество векторов из  $N$  одинаковых чисел, принимающих значения из множества  $\Omega$  обозначим как  $\bar{\Omega} \subset \Omega^N$ , т.е.  $\bar{\Omega} = ((j, j, \dots, j), j \in \Omega)$ . Всего таких векторов  $m$ .

Согласно (2.23), если  $q = 1$ , то вероятность того, что все вершины последовательности находятся в одном и том же состоянии  $j$ , где  $j \in \Omega$  равна:

$$\begin{aligned} \zeta(X = \mathbf{j}) &= p_{t_1}(x_{t_1} = j) \prod_{i=2}^N q_{t_i}(x_{t_i} = j | x_{t_{i-1}} = j) = \\ &= p_{t_1}(x_{t_1} = j), \text{ где } \mathbf{j} = (j, j, \dots, j) \in \bar{\Omega}. \end{aligned} \quad (2.24)$$

С другой стороны, в частной модели марковского случайного поля маргинальные априорные распределения во всех элементах являются равномерными (1.19), т.е.  $q_{t_i}(x_{t_i}) = \frac{1}{m}; x_{t_i} \in \Omega; i = 1, \dots, N$ .

$$\text{Поэтому } \zeta(X = \mathbf{j}) = \frac{1}{m}, \text{ где } \mathbf{j} \in \bar{\Omega}. \quad (2.25)$$

Согласно (2.23), также легко увидеть, что вероятность появления любой другой последовательности, вершины которой находятся в разных состояниях, равна 0, т.е.  $\zeta(X = \mathbf{j}) = 0$ , где  $\mathbf{j} \in \Omega^N, \mathbf{j} \notin \bar{\Omega}$ .

Очевидно, что вероятность появления последовательности скрытых одинаковых состояний любой длины равна

$$\zeta(X \in \bar{\Omega}) = \sum_{j \in \bar{\Omega}} \zeta(X = j) = m \frac{1}{m} = 1. \quad (2.26)$$

Выражение (2.25) означает, что вероятность появления последовательности скрытых одинаковых состояний  $j \in \Omega$  любой длины  $N \geq 2$  равна  $\frac{1}{m}$ .

Поэтому при  $q = 1$  скрытая марковская модель является вырожденной и никак не влияет на качество результата распознавания. Результат распознавания полностью опирается на комбинации апостериорных распределений, полученных на этапе независимого обучения.

Если распределения  $p_s(x_s | y_s), s \in T_t^+, t \in T$  резко неравномерны (хорошее качество независимого обучения), то форма (2.14) начального распределения  $p_t(x_t = i | Y_t^+), t = t_1$  и далее  $t = t_2, t_3, \dots$  поддерживается при  $x_t = i$  на объектах этого же класса и немедленно начинает выравниваться при появлении серии отчетов другого класса  $x_t \neq i$ .

Если последовательность подряд идущих объектов другого класса  $x_t \neq i$  оказывается достаточно длинной, то распределения  $p_t(x_t = i | Y_t^+)$  снова становятся резко неравномерными, но в пользу уже другого класса  $x_t \neq i$  и т.д.

Если распределения  $p_s(x_s | y_s)$  более равномерны (плохое качество независимого обучения), то, в целом, они примерно и остаются такими же.

Аналогично можно описать характер изменения распределений  $p_s(x_s = i | Y)$ , согласно (2.19).

Таким образом, результаты распознавания при предельных значениях диагонального элемента ( $q = 1/m$  и  $q = 1$ ) оказываются хуже, чем при промежуточных случаях когда  $1/m < q < 1$ .

### 2.3 Алгоритм подбора диагонального элемента для ациклического графа, основанный на независимом обучении

На основе формулы (2.9) мы предложим алгоритм подбора диагонального элемента, который заключается в итерационном решении двух взаимосвязанных задач:

- а) При фиксированных  $q$  и  $Y$  решить задачу (2.2).
- б) При фиксированных  $X$  и  $Y$  решить задачу (2.8).

Данный алгоритм подбора диагонального элемента состоит из следующих шагов:

1. Выбрать ациклический граф соседства из заранее заданного набора.
2. На основе результата независимого обучения, оценить значение диагонального элемента:

$$q = \frac{|V_1|}{|T|-1}, \text{ где } V_1 = \{u \in T_{(t^*)} \mid x_u = x_v, v \in T_{(u)}^{-0}\}.$$

3. Однократным применением базового алгоритма распознавания с пересчитанным диагональным элементом перейти от распределений  $p_t(x_t | y_t)$ , полученных ранее для независимого обучения, к апостериорным распределениям  $p_t(x_t | Y)$ ,  $t \in T$ , соответствующим реализации  $X$  при наблюдении  $Y$ .

4. Оценить значение диагонального элемента:

$$q = \frac{|V_1|}{|T|-1}, \text{ где } V_1 = \{u \in T_{(t^*)} \mid x_u = x_v, v \in T_{(u)}^{-0}\}.$$

5. Вновь применить алгоритм распознавания с пересчитанным диагональным элементом  $q$ , где вместо распределений  $p_t(x_t | y_t)$  рассмотреть только что полученные распределения  $p_t(x_t | Y)$ , и перейти к новым апостериорным распределениям, которые снова обозначить как  $p_t(x_t | Y)$ ,  $t \in T$ . Снова оценить диагональный элемент  $q$ .

6. Повторять шаг 5 до тех пор, пока решения  $\hat{x}_t = \arg \max_{x_t \in \Omega} p_t(x_t | Y)$  не перестанут меняться или не начнут циклически повторяться.

## 2.4 Алгоритм подбора диагонального элемента для ациклического графа, начиная с заранее заданного значения

В предыдущем алгоритме на втором шаге значение диагонального элемента пересчитывается на основе результата независимого обучения. Но независимое обучение часто дает плохой результат. Рассмотрим другой алгоритм, который отличается от предыдущего тем, что на втором шаге значение диагонального шага задается эвристически.

Новый алгоритм состоит из следующих шагов:

1. Выбрать ациклический граф соседства из заранее заданного набора.
2. Задать значение диагонального элемента, например,  $q = 0.95$ .
3. Однократным применением базового алгоритма распознавания с пересчитанным диагональным элементом перейти от распределений  $p_t(x_t | y_t)$ , полученных ранее для независимого обучения, к апостериорным распределениям  $p_t(x_t | Y)$ ,  $t \in T$ , соответствующим реализации  $X$  при наблюдении  $Y$ .

4. Оценить значение диагонального элемента:

$$q = \frac{|V_1|}{|T|-1}, \text{ где } V_1 = \{u \in T_{(t^*)} \mid x_u = x_v, v \in T_{(u)}^{-0}\}.$$

5. Вновь применить алгоритм распознавания с пересчитанным диагональным элементом  $q$ , где вместо распределений  $p_t(x_t | y_t)$  рассмотреть только что полученные распределения  $p_t(x_t | Y)$ , и перейти к новым апостериорным распределениям, которые снова обозначить как  $p_t(x_t | Y)$ ,  $t \in T$ . Снова оценить диагональный элемент  $q$ .

6. Повторять шаг 5 до тех пор, пока решения  $\hat{x}_t = \arg \max_{x_t \in \Omega} p_t(x_t | Y)$  не перестанут меняться или начнут циклически повторяться.

## 2.5 Алгоритм подбора диагонального элемента для ациклического графа по схеме Гаусса-Зайделя

Одним из популярных методов оптимизации многомерных функций является метод покоординатного спуска Гаусса-Зайделя. Ранее [11,12] данная схема была рассмотрена с целью подбора весов комбинации ациклических

графов соседства, где варьирование веса графа рассматривалось как координатное варьирование. Здесь задача оптимизации вырождается, т.к. задан только один ациклический граф соседства, а искомая функция зависит от только одной переменной  $q$ .

На основе схемы Гаусса-Зайделя предложим другой алгоритм подбора диагонального элемента, состоящий из следующих шагов:

1. Выбрать ациклический граф соседства из заранее заданного набора.
2. Варьировать значение  $q$  в диапазоне от  $1/m$  до 1. Для каждого значения  $q$  применить базовый алгоритм итерационно до тех пор, пока число ошибок не перестанет изменяться, и оценить число ошибок.
3. Найти значение  $q$ , которое обеспечило минимальное число ошибок.



### 3 ЗАДАЧА ПОДБОРА ПАРАМЕТРОВ КОМБИНИРОВАНИЯ АЦИКЛИЧЕСКИХ ГРАФОВ СОСЕДСТВА

Как было показано, для улучшения качества распознавания нужно найти подходящее значение диагонального элемента матрицы вероятностей условных вероятностей переходов. Поэтому естественным образом возникает задача одновременного подбора диагонального элемента и весов графов в их линейной комбинации. Для решения данной задачи следует включить шаги варьирования диагонального элемента в предложенную ранее [11, 12] схему Гаусса-Зайделя при подборе весов графов. Для этого сначала проведем предварительные эксперименты для исследования свойств предложенного в [11, 12] алгоритма подбора весов.

#### 3.1 Свойства алгоритма подбора весов ациклических графов в их линейной комбинации

В алгоритме подбора весов графов [11, 12] при оценке числа ошибок применяется однократное распознавание. Рассмотрим свойства этого алгоритма при однократном и многократном распознавании.

Для этого рассмотрим следующее трехклассовое изображение (рис.3.1).

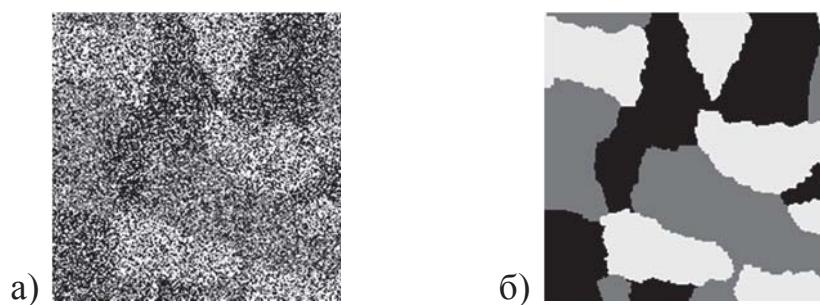


Рисунок 3.1 – Исходные данные: а) Текстурное изображение;  
б) Данные о точной сегментации

На данном изображении для диагонального элемента  $q = 0,9$ , были получены следующие веса, соответствующие пяти заданным ациклическим графам (рис. 3.2).

Эти веса доставляют минимум целевой функции, которая в данном случае является числом ошибок.

Попробуем совместное варьирование весов ациклических графов и диагонального элемента вокруг точки минимума.

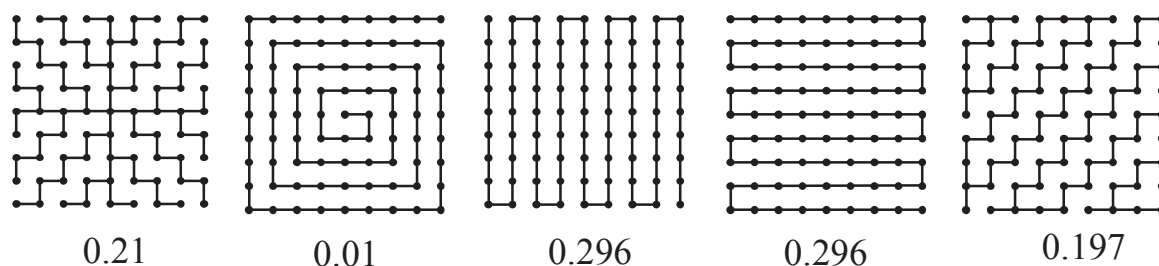


Рисунок 3.2 – Ациклические графы соседства и их веса

Рассмотрим последовательно каждый ациклический граф. Для каждого графа сначала берем диагональный элемент и его варьлируем от  $1/3=0.33$  до 1. Для каждого значения диагонального элемента будем варьировать вес данного ациклического графа от 0 до 1, масштабируя веса остальных графов. Причем для каждого значения диагонального элемента и набора весов сначала применим однократное распознавание (один шаг алгоритма комбинирования ациклических графов), а в другой серии экспериментов – многократное распознавание (итерационный алгоритм комбинирования ациклических графов). В результате получим множество значений ошибок, зависящих от веса рассмотренного графа и значения диагонального элемента.

### 3.1.1 Предварительные эксперименты в случае однократного распознавания

В случае однократного распознавания графики зависимостей числа ошибок от веса каждого графа при разных значениях диагонального элемента приведены на рис. 3.3–3.7.

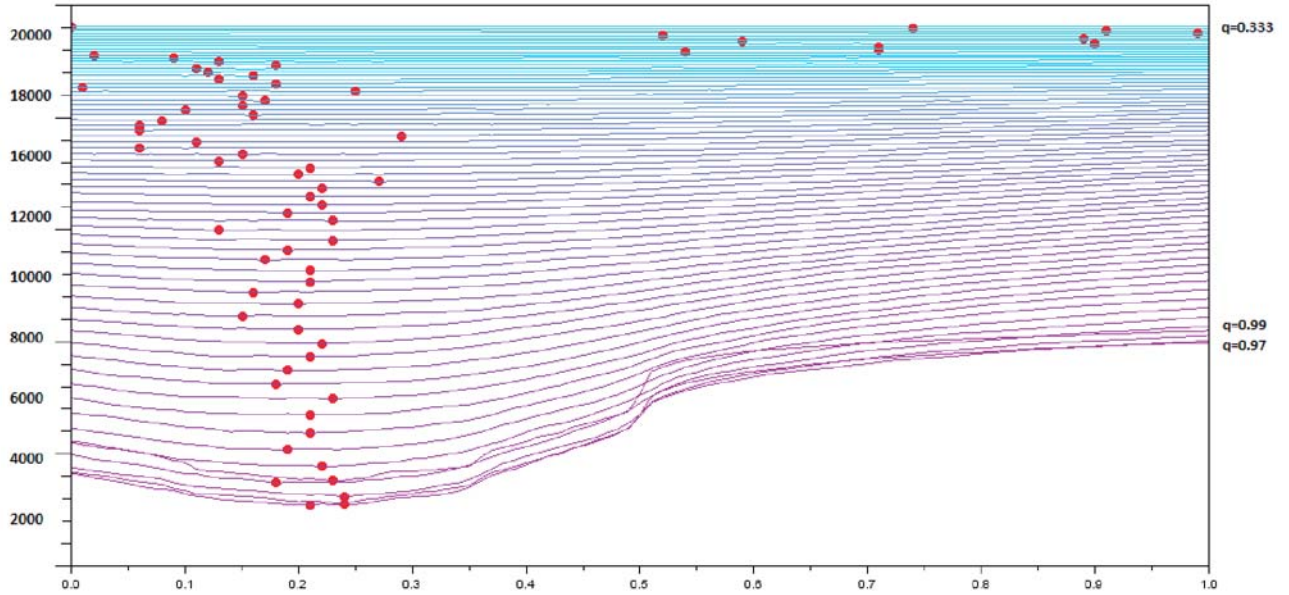


Рисунок 3.3 – Графики зависимости числа ошибок от веса первого графа при разных значениях диагонального элемента

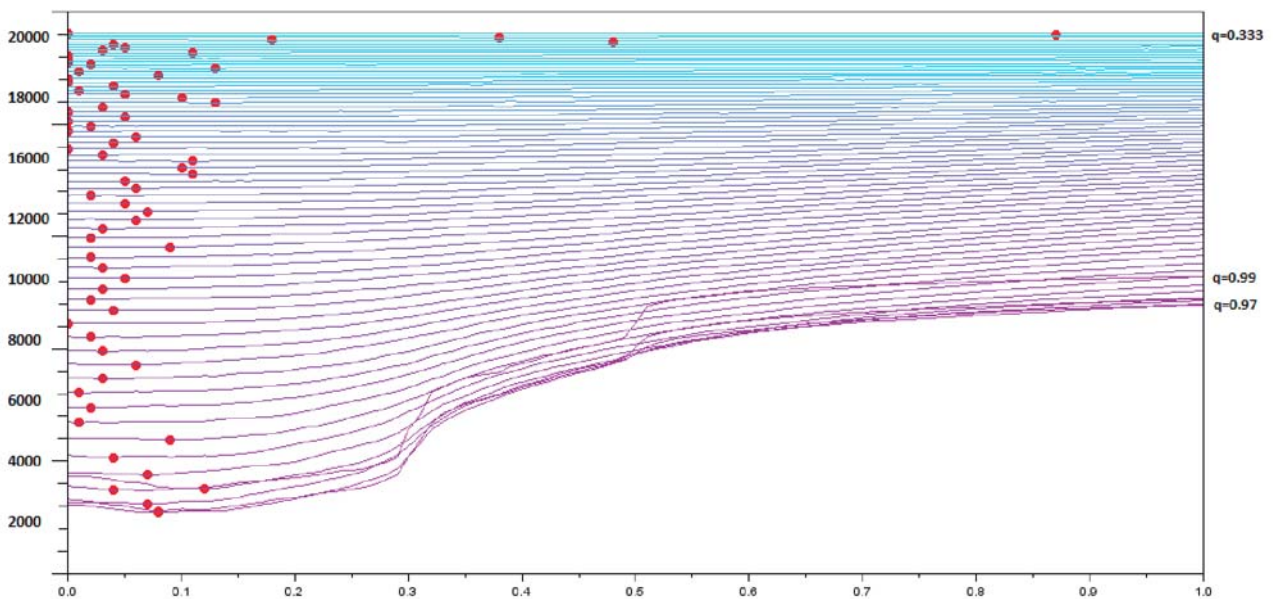


Рисунок 3.4 – Графики зависимости числа ошибок от веса второго графа при разных значениях диагонального элемента

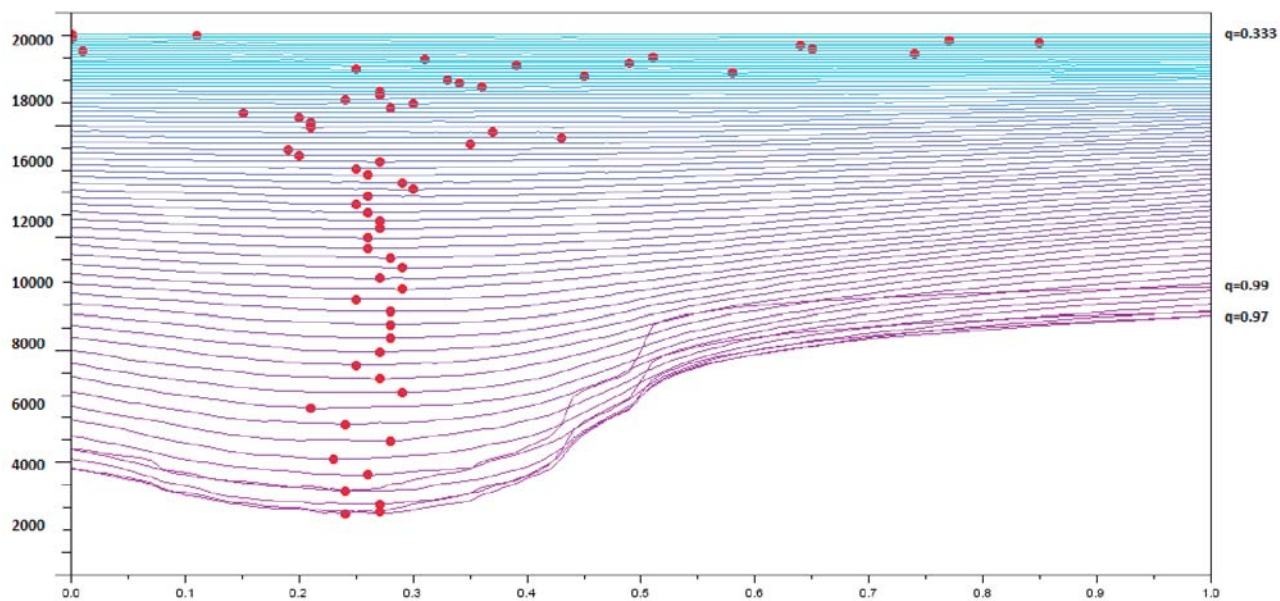


Рисунок 3.5 – Графики зависимости числа ошибок от веса третьего графа при разных значениях диагонального элемента

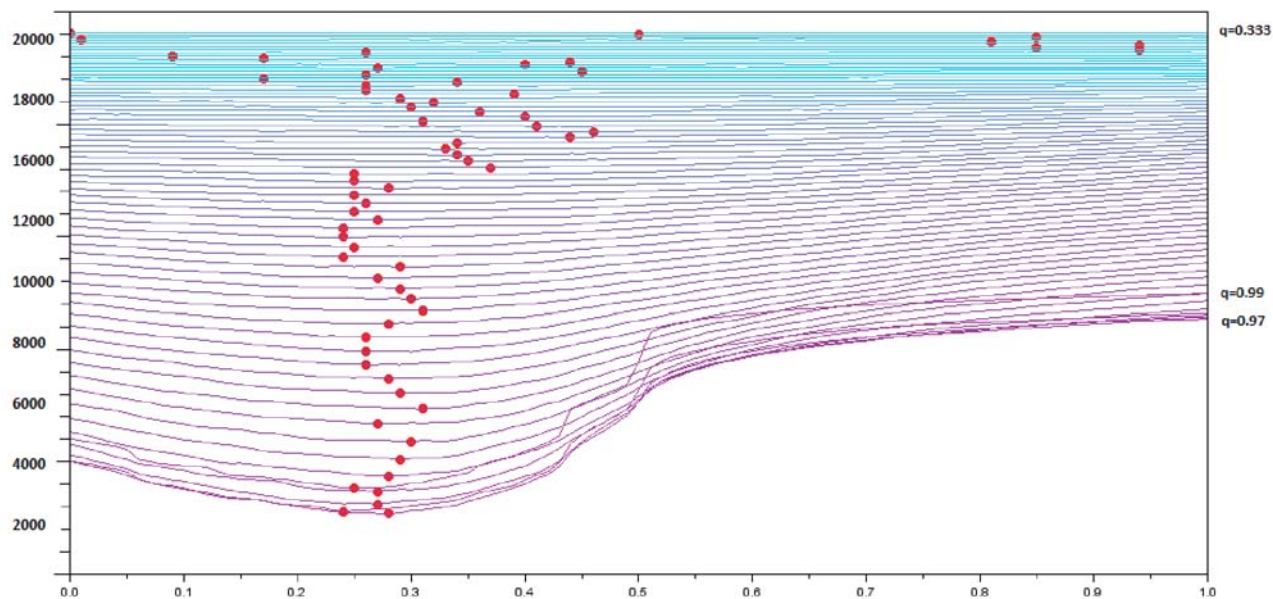


Рисунок 3.6 – Графики зависимости числа ошибок от веса четвертого графа при разных значениях диагонального элемента

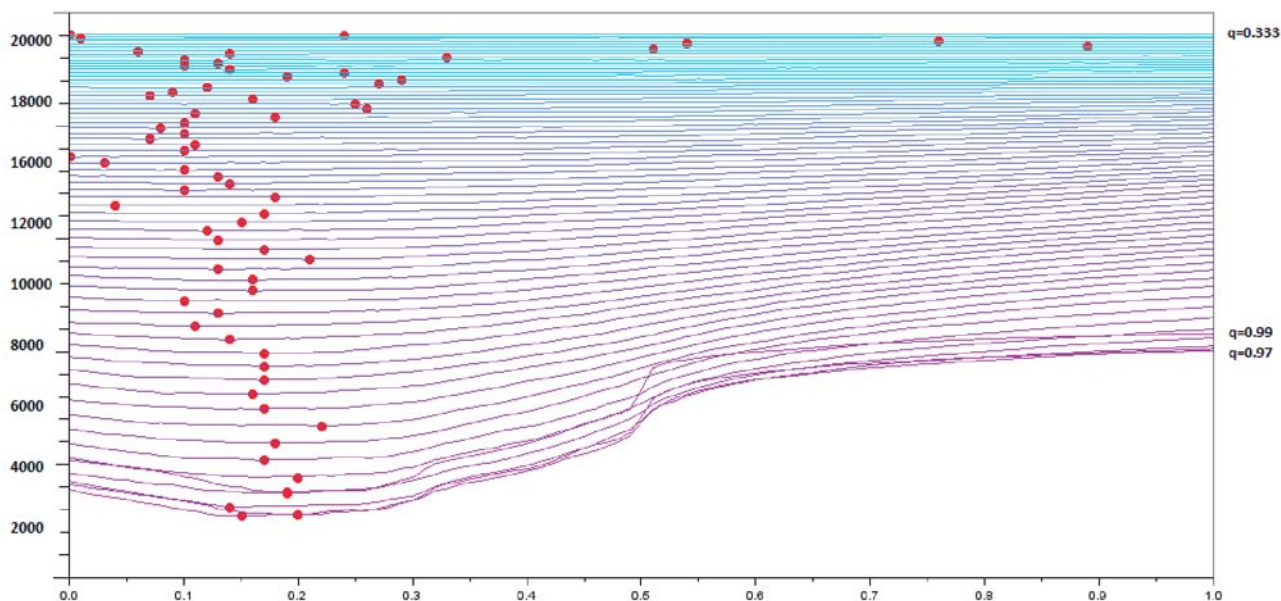


Рисунок 3.7 – Графики зависимости числа ошибок от веса пятого графа при разных значениях диагонального элемента

Таким образом, в случае однократного распознавания линии числа ошибок не пересекаются, и увеличение диагонального элемента ведет к снижению числа ошибок при фактически том же весе графа, которые мы до этого нашли. Вес является оптимальным и фактически не меняется.

Увеличение диагонального элемента приводит к снижению числа ошибок до какого-то момента, т.к. диагональный элемент не может быть равен 1. Поэтому функция зависимости числа ошибок от значения диагонального элемента должна иметь перегиб. Чтобы увидеть, где находятся точки перегиба, посмотрим эти линии ошибок в другой плоскости (рис. 3.8–3.12).

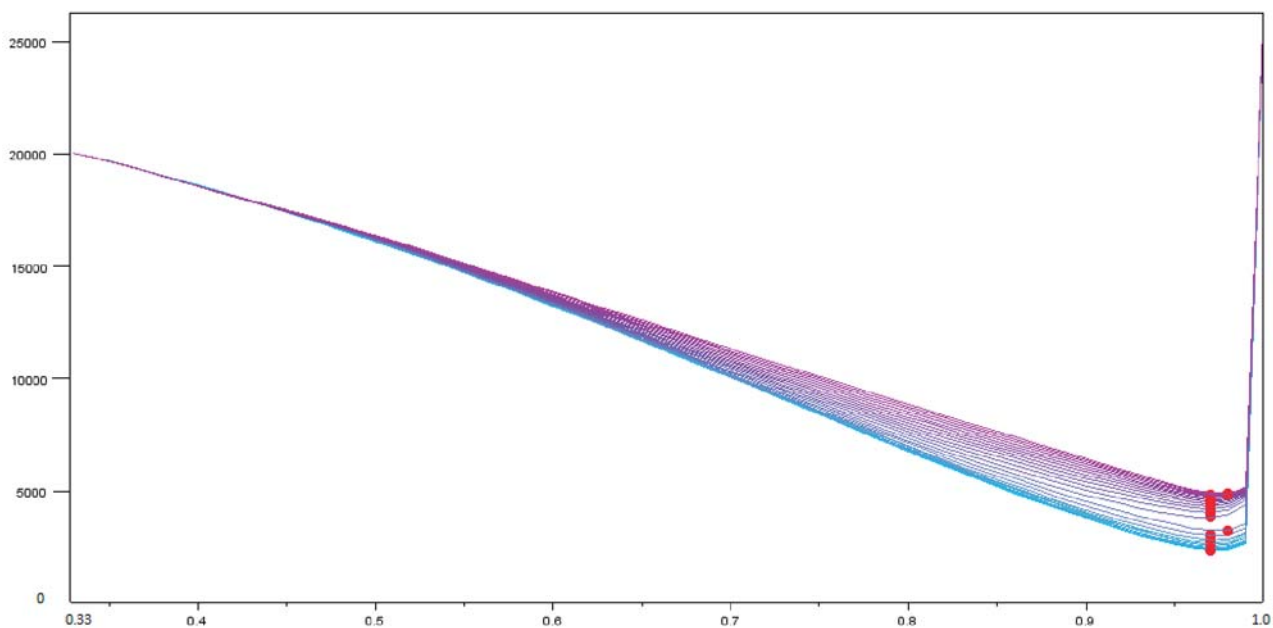


Рисунок 3.8 – Графики зависимости числа ошибок от значения диагонального элемента при разных значениях веса первого графа

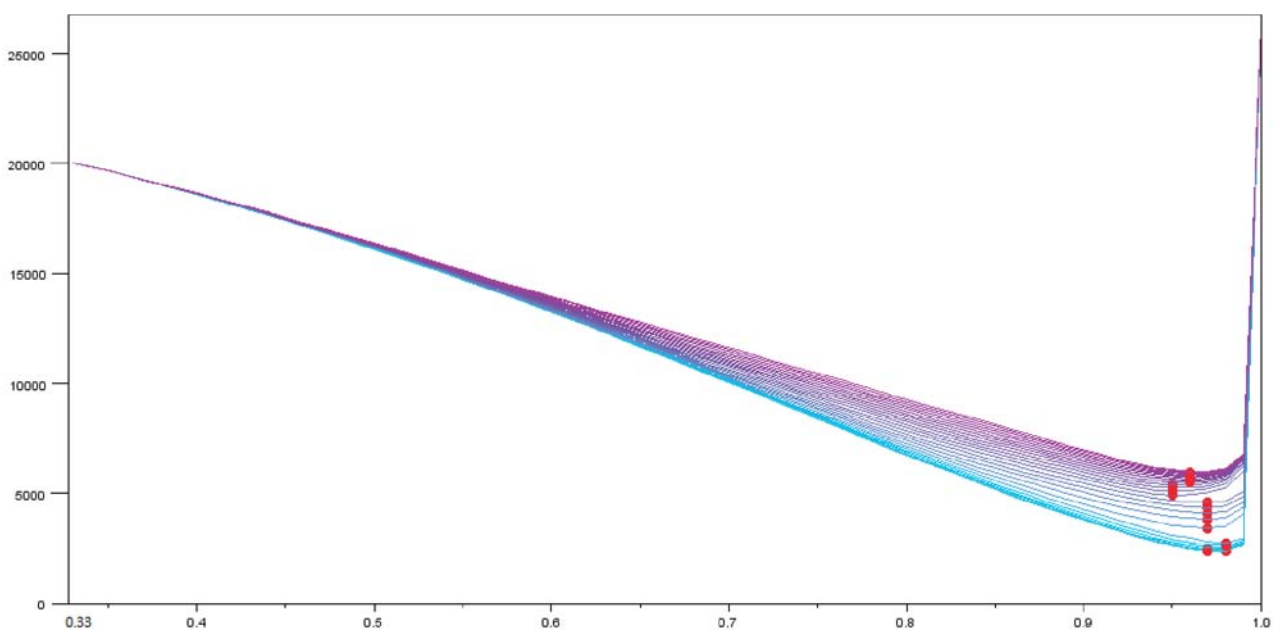


Рисунок 3.9 – Графики зависимости числа ошибок от значения диагонального элемента при разных значениях веса второго графа

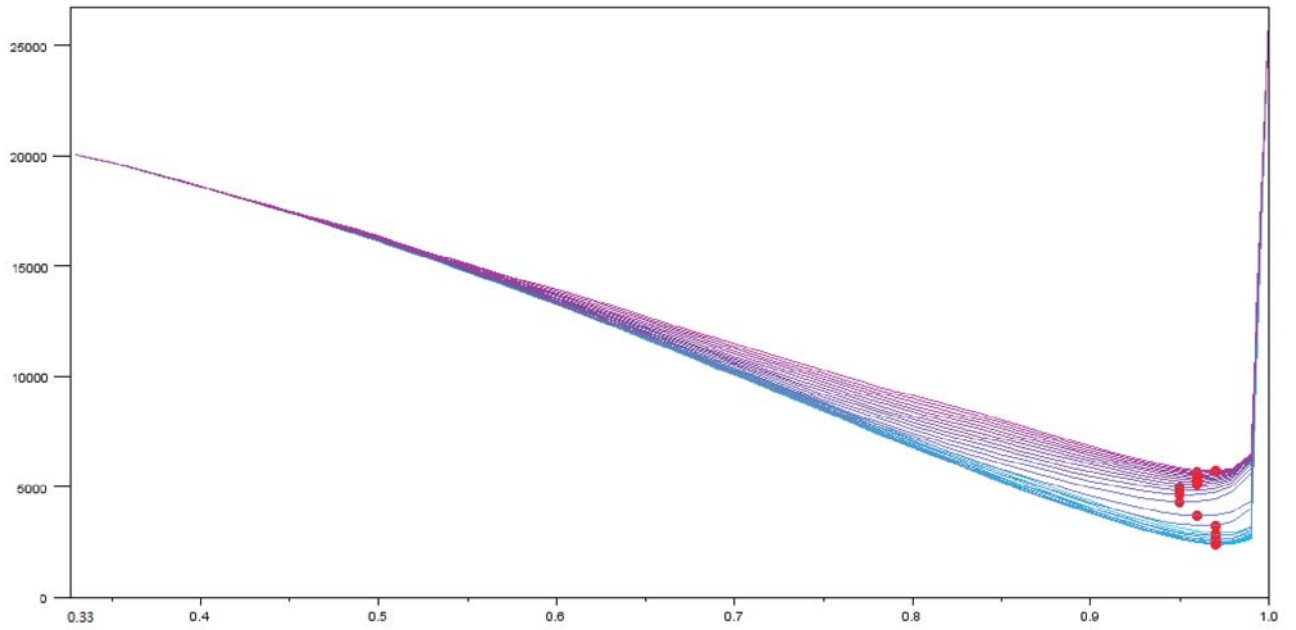


Рисунок 3.10 – Графики зависимости числа ошибок от значения диагонального элемента при разных значениях веса третьего графа

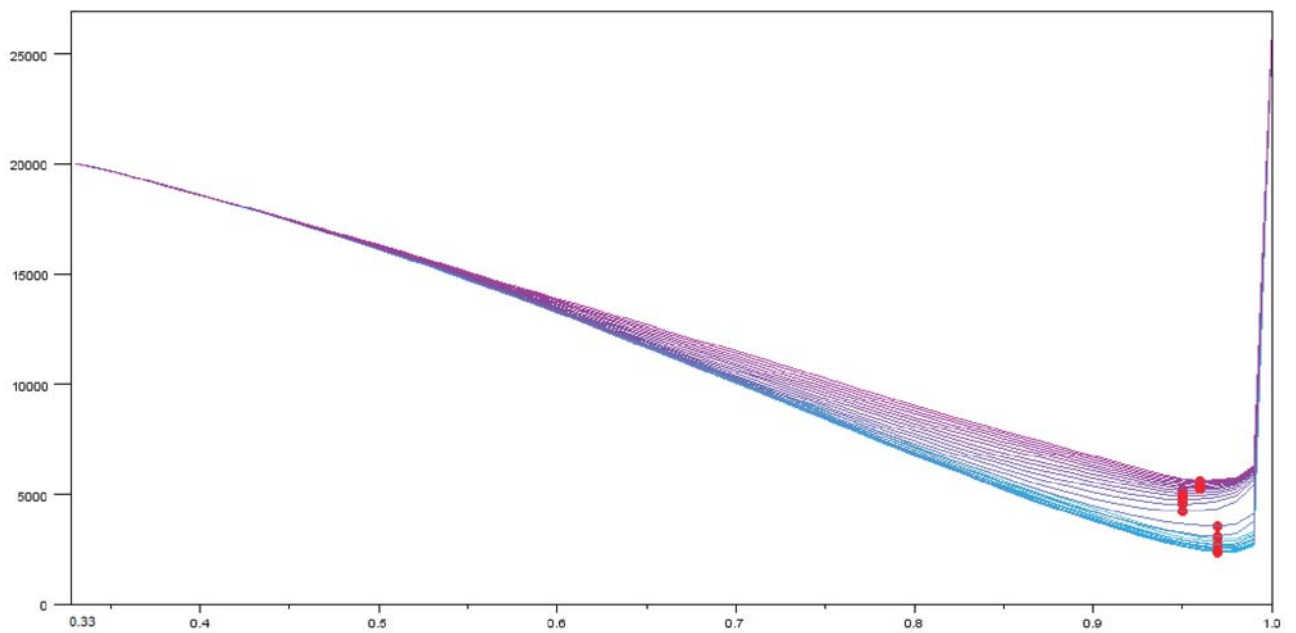


Рисунок 3.11 – Графики зависимости числа ошибок от значения диагонального элемента при разных значениях веса четвертого графа

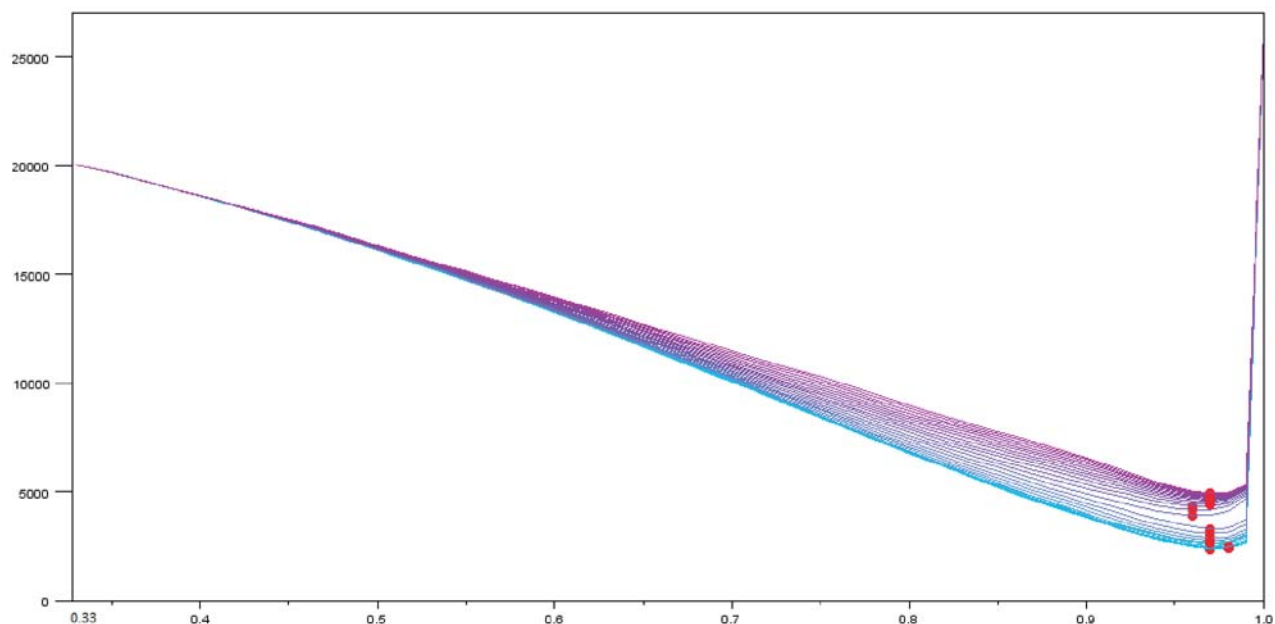


Рисунок 3.12 – Графики зависимости числа ошибок от значения диагонального элемента при разных значениях веса пятого графа

Таким образом, в случае однократного распознавания оптимальное значение диагонального элемента, которое обеспечивает минимальное число ошибок, всегда приближается к единице и примерно находится в диапазоне  $[0.95; 0.98]$ .

В итоге, задача поиска значения диагонального элемента оказывается вырожденной. Это означает, что нужно всегда просто сделать диагональный элемент как можно ближе к единице.

### 3.1.2 Предварительные эксперименты в случае многократного распознавания

В случае многократного распознавания графики зависимости числа ошибок от веса каждого графа при разных значениях диагонального элемента приведены на рис. 3.13–3.17.



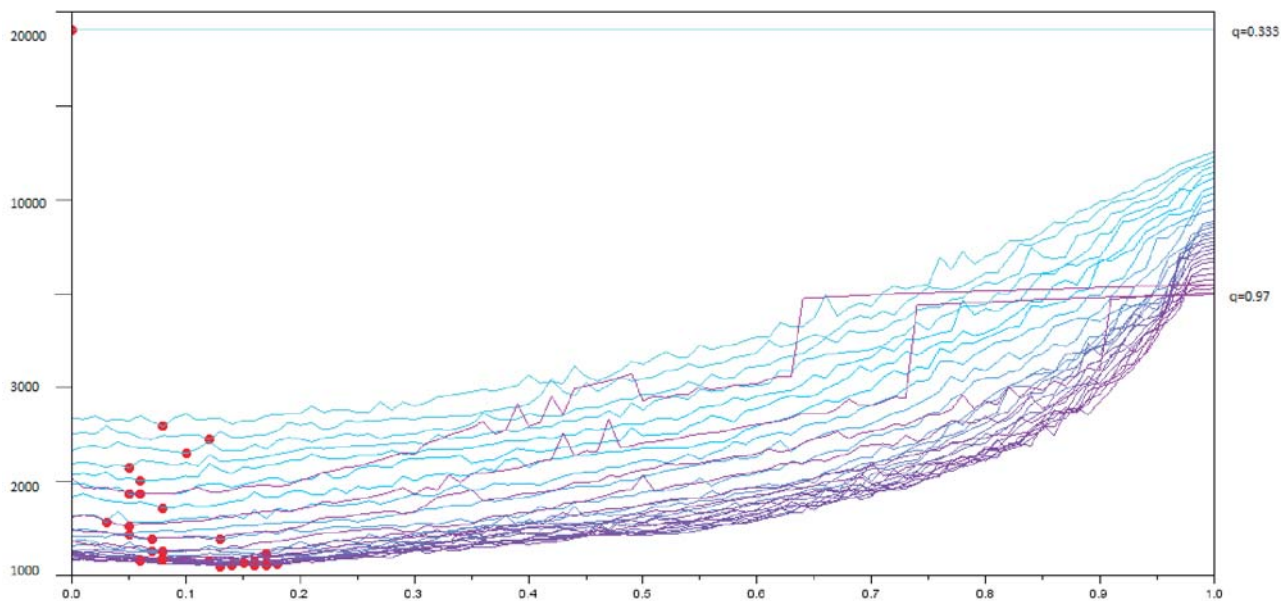


Рисунок 3.13 – Графики зависимости числа ошибок от веса первого графа при разных значениях диагонального элемента

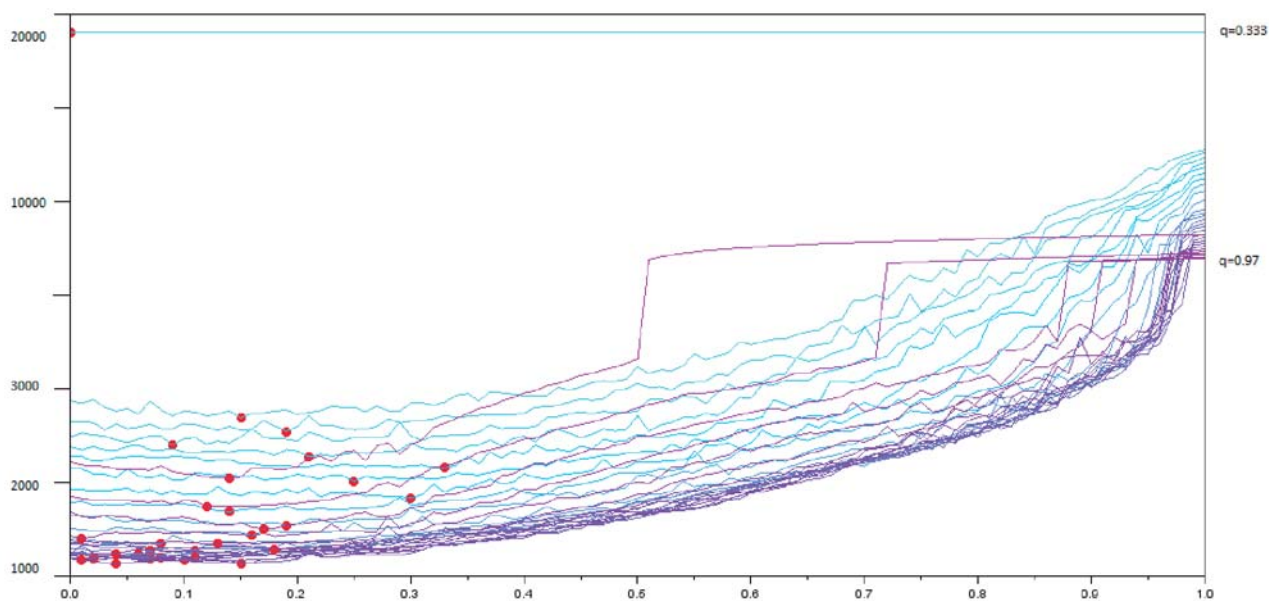


Рисунок 3.14 – Графики зависимости числа ошибок от веса второго графа при разных значениях диагонального элемента

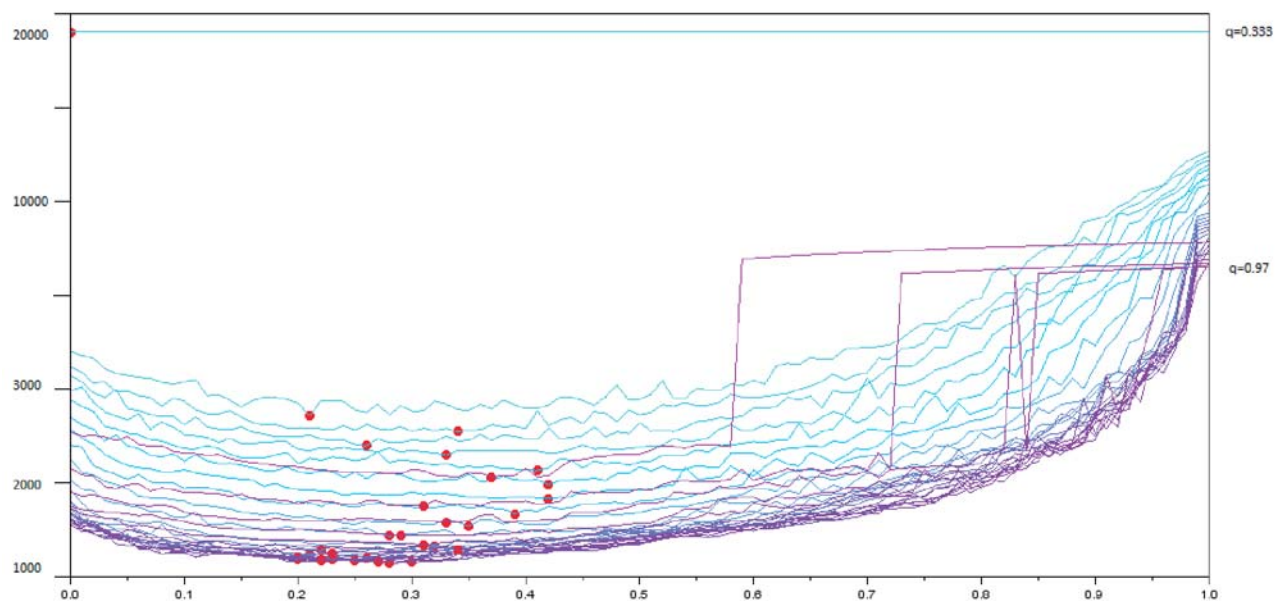


Рисунок 3.15 – Графики зависимости числа ошибок от веса третьего графа при разных значениях диагонального элемента

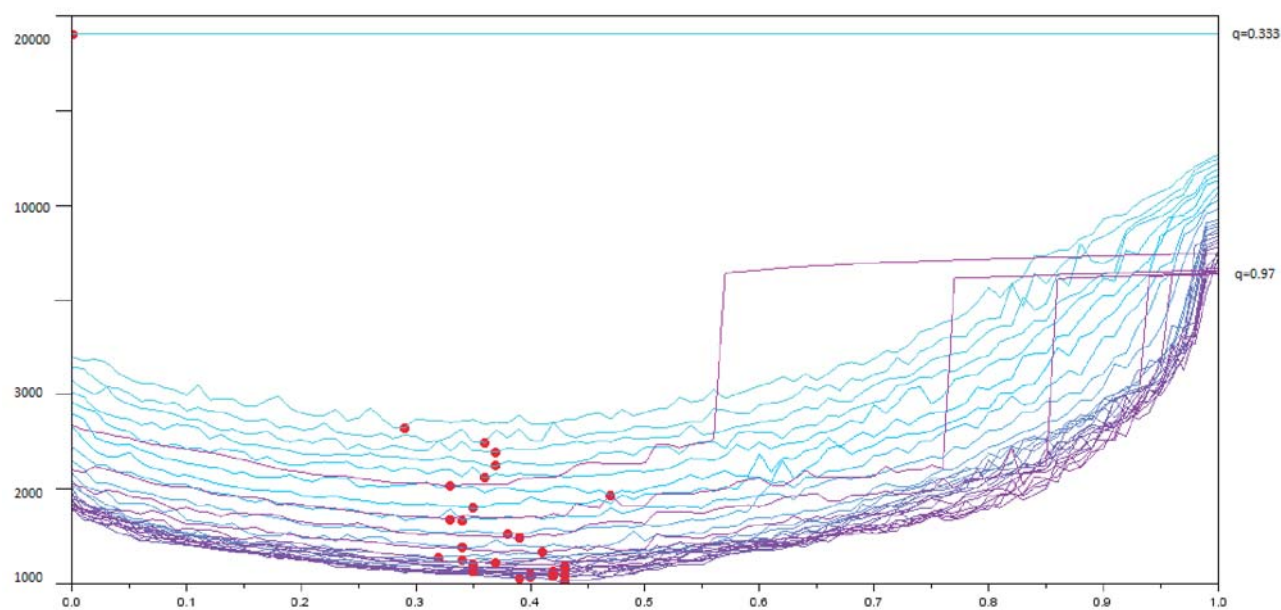


Рисунок 3.16 – Графики зависимости числа ошибок от веса четвертого графа при разных значениях диагонального элемента

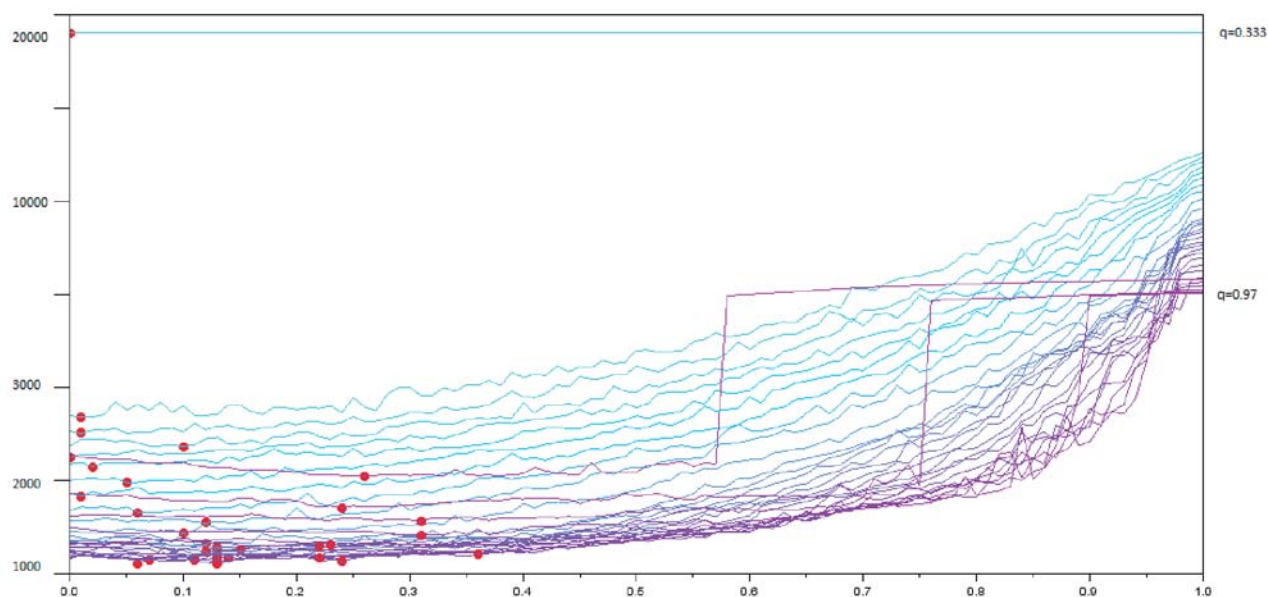


Рисунок 3.17 – Графики зависимости числа ошибок от веса пятого графа при разных значениях диагонального элемента

В случае многократного распознавания при увеличении диагонального элемента линии ошибок оказываются неплавными и пересекающимися. В итоге, задача поиска минимума ошибок при варьировании диагонального элемента оказывается невырожденной.

Таким образом, на основе предварительных экспериментов можно сделать следующие выводы:

- В случае однократного распознавания линии числа ошибок не пересекаются. Увеличение диагонального элемента ведет к снижению числа ошибок фактически при том же весе графа, который мы до этого нашли. Вес графа оказывается оптимальным.

- Увеличение диагонального элемента приводит к снижению числа ошибок до какого-то момента, т.к. диагональный элемент не может быть равен единице. В этом случае оптимальное значение диагонального элемента, которое соответствует точке перегиба и обеспечивает минимальное число ошибок, всегда приближается к единице и примерно находится в диапазоне  $[0.95; 0.98]$ . Задача поиска вырождена. Это означает, что находясь в точке ми-

нимума, нужно просто сделать диагональный элемент как можно ближе к единице.

- В случае многократного распознавания при увеличении диагонального элемента линии ошибок будут неплавными и пересекающимися. Вес графа будет меняться, т.е. мы не будем оставаться в указанной точке минимума. В итоге, оказывается, что разные комбинации весов графов и значений диагонального элемента соответствуют разному числу ошибок распознавания.

Расширим схему Гаусса-Зайделя для поиска оптимального значения диагонального элемента матрицы переходов. При однократном распознавании новый шаг варьирования диагонального элемента в диапазоне  $1/m \leq q < 1$  оказывается вырожденным, т. к. при любом наборе весов ациклических графов монотонное увеличение диагонального элемента при условии  $q < 1$  монотонно уменьшает число ошибок распознавания. Чтобы задача поиска диагонального элемента не оказалась вырожденной, в новых алгоритмах подбора параметров комбинирования графов будем применять многократное распознавание при оценке числа ошибок.

## **3.2 Алгоритмы подбора параметров комбинирования ациклических графов соседства**

### **3.2.1 Алгоритм подбора единственного диагонального элемента и весов ациклических графов соседства**

В этом алгоритме считается, что все ациклические марковские модели, соответствующие графам  $G_k, k = 1, \dots, K$ , определяются одним общим диагональным элементом  $q$ . Сначала все веса одинаковы  $w^* = \{w_i = 1/K, i = 1, \dots, K\}$ . Суть данного алгоритма – это включение поиска значения диагонального элемента в схему Гаусса-Зайделя при подборе весов графов. Изменение веса графа в диапазоне от 0 до 1 и изменение диагонального элемента в диапазоне от  $1/m$  до 1 считаются аналогом по координатного варьирования. Мы поочередно варьируем веса графов и находим тот граф и

его вес, который обеспечил наименьшее число ошибок в совокупности с весами других графов. Потом варьируем диагональный элемент и находим его оптимальное значение. Процесс повторяется до тех пор, пока число ошибок не перестанет меняться.

Этот алгоритм состоит из следующих шагов:

1. Проварьируем  $q$  и найдем его значение при минимальном числе ошибок  $E$ :  $q^* = \arg \min_{1/m \leq q < 1} E(\mathbf{w}^*, q)$ .

2. Шаг варьирования весов по всем графам. Варьируется вес  $w_k$  очередного графа  $G_k$  в диапазоне  $0 \leq w_k \leq 1$  при масштабировании весов остальных графов. Каждое пробное варьирование проверяется многократным распознаванием с подсчетом числа ошибок  $E(\mathbf{w}, q^*)$ . Для каждого графа  $G_k$  определим число ошибок и вектор весов  $\mathbf{w}_k^* = (w_1, \dots, w_k^*, \dots, w_K)$ :

$$E_k^* = \min_{0 \leq w_k \leq 1} E(\mathbf{w}, q^*), \quad \mathbf{w}_k^* = \arg \min_{0 \leq w_k \leq 1} E(\mathbf{w}, q^*)$$

3. Среди всех наборов  $\mathbf{w}_k^*$ ,  $k = 1, \dots, K$  найдем набор  $\mathbf{w}^* = \mathbf{w}_{k^*}^*$ ,  $k^* = \arg \min_{1 \leq k \leq K} E_k^*$ , обеспечивший наименьшее число ошибок.

4. Повторим шаги 1–3 до тех пор, пока число ошибок распознавания не перестанет изменяться.

### 3.2.2 Первая схема подбора диагональных элементов и весов ациклических графов соседства

В отличие от предыдущего алгоритма здесь считается, что каждый ациклический граф соответствует отдельной марковской цепи, которая определяется своей матрицей вероятностей условных вероятностей переходов, заданной одним значением ее диагонального элемента. Это значит, что каждому графу  $G_k$ ,  $k = 1, \dots, K$  соответствует отдельная ациклическая марковская модель со своим диагональным элементом  $q_k$ ,  $k = 1, \dots, K$ . Сначала все веса также считаются одинаковыми  $\mathbf{w}^* = \{w_i = 1/K, i = 1, \dots, K\}$ . В первой схеме

варьирования параметров сначала мы варьируем веса всех графов, а потом диагональные элементы.

Первая схема алгоритма подбора диагональных элементов и весов графов состоит из следующих шагов:

1. Проварьируем одновременно все диагональные элементы  $q_k = q, k = 1, \dots, K$  и найдем значение  $q^*$  при минимальном числе ошибок  $E$ :

$$\mathbf{q}^* = (q^*, \dots, q^*) = \arg \min_{1/m \leq q < 1} E(\mathbf{w}^*, \mathbf{q}).$$

2. Шаг варьирования весов по всем графам. Варьируется вес  $w_k$  очередного графа  $G_k$  в диапазоне  $0 \leq w_k \leq 1$  при масштабировании весов остальных графов. Каждое пробное варьирование проверяется многократным распознаванием с подсчетом числа ошибок  $E(\mathbf{w}, \mathbf{q}^*)$ . Для каждого графа  $G_k$  определим число ошибок и вектор весов  $\mathbf{w}_k^* = (w_1, \dots, w_k^*, \dots, w_K)$ :

$$E_k^* = \min_{0 \leq w_k \leq 1} E(\mathbf{w}, \mathbf{q}^*), \quad \mathbf{w}_k^* = \arg \min_{0 \leq w_k \leq 1} E(\mathbf{w}, \mathbf{q}^*).$$

3. Среди всех наборов  $\mathbf{w}_k^*, k = 1, \dots, K$  найдем набор  $\mathbf{w}^* = \mathbf{w}_{k^*}^*, k^* = \arg \min_{1 \leq k \leq K} E_k^*$ , обеспечивший наименьшее число ошибок.

4. Шаг варьирования всех диагональных элементов. В диапазоне  $1/m \leq q_k < 1$  варьируется диагональный элемент  $q_k$ , соответствующий графу  $G_k$ . Остальным графам соответствуют элементы  $q_i, i = 1, \dots, K, i \neq k$ , найденные до варьирования  $q_k$ , которые остаются постоянными. Найдем значение  $q_k$  при минимальном числе ошибок:

$$q_k^* = \arg \min_{1/m \leq q_k < 1} E(\mathbf{w}^*, \mathbf{q}).$$

Новое значение  $q_k^*$  применяется при варьировании остальных диагональных элементов  $q_{k+1}, q_{k+2}, \dots, q_K$ . В итоге, получим оптимальный вектор  $\mathbf{q}^*$ .

5. Повторим шаги 2–4 до тех пор, пока число ошибок распознавания не перестанет изменяться.

### 3.2.3 Вторая схема подбора диагональных элементов и весов ациклических графов соседства

Каждому графу  $G_k, k=1...K$  соответствует отдельная ациклическая марковская модель со своим диагональным элементом  $q_k, k=1...K$ . Сначала все веса также считаются одинаковыми  $\mathbf{w}^* = \{w_i = 1/K, i=1, \dots, K\}$ . Отличие от второго алгоритма заключается в порядке варьирования параметров. Здесь мы варьируем вес каждого графа и потом сразу соответствующий ему диагональный элемент.

1. Проварьируем одновременно все диагональные элементы  $q_k = q, k=1, \dots, K$  и найдем значение  $q^*$  при минимальном числе ошибок  $E$ :

$$\mathbf{q}^* = (q^*, \dots, q^*) = \arg \min_{1/m \leq q < 1} E(\mathbf{w}^*, \mathbf{q}).$$

2. Шаг варьирования весов и диагональных элементов по всем графам. Варьируется вес  $w_k$  очередного графа  $G_k$  в диапазоне  $0 \leq w_k \leq 1$  при масштабировании весов остальных графов. Каждое пробное варьирование проверяется многократным распознаванием с подсчетом числа ошибок  $E(\mathbf{w}, \mathbf{q}^*)$ . Для каждого графа  $G_k$  определим число ошибок и вектор весов  $\mathbf{w}_k^* = (w_1, \dots, w_k^*, \dots, w_K)$ :

$$E_k^* = \min_{0 \leq w_k \leq 1} E(\mathbf{w}, \mathbf{q}^*), \quad \mathbf{w}_k^* = \arg \min_{0 \leq w_k \leq 1} E(\mathbf{w}, \mathbf{q}^*).$$

Далее варьируется элемент  $q_k$ , соответствующий графу  $G_k$ , в диапазоне  $1/m \leq q_k < 1$ . Остальным графам соответствуют элементы  $q_i, i=1, \dots, K, i \neq k$ , найденные до варьирования  $q_k$ , которые остаются постоянными. Найдем значение  $q_k$  при минимальном числе ошибок:

$$q_k^* = \arg \min_{1/m \leq q_k < 1} E(\mathbf{w}_k^*, \mathbf{q}).$$

Значение  $q_k^*$  определяет вектор  $\mathbf{q}_k^* = (q_1, \dots, q_k^*, \dots, q_K)$ .

3. Среди полученных пар  $(\mathbf{w}_k^*, \mathbf{q}_k^*), k=1, \dots, K$  найдем пару

$(\mathbf{w}^*, \mathbf{q}^*) = (\mathbf{w}_{k^*}^*, \mathbf{q}_{k^*}^*), k^* = \arg \min_{1 \leq k \leq K} E(\mathbf{w}_k^*, \mathbf{q}_k^*)$ , обеспечившую наименьшее число ошибок.

4. Повторим шаги 2–3 до тех пор, пока число ошибок распознавания не перестанет изменяться.

### **3.2.4 Сходимость алгоритмов подбора параметров комбинирования ациклических графов соседства**

Заметим, что точка экстремума, полученная на одном шаге, всегда входит в диапазон варьирования параметров на следующем шаге. Это гарантирует монотонное убывание числа ошибок  $E^*$  по шагам варьирования.

С другой стороны, число ошибок  $E^*$  должно быть натуральным. Поэтому последовательность значений пошаговых наименьших чисел ошибок должна сходиться. Т.е. всегда существует момент, когда число ошибок  $E^*$  перестанет меняться. Таким образом, все построенные алгоритмы подбора параметров комбинирования ациклических графов сходятся.



## 4 ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ АЛГОРИТМОВ

### 4.1 Модельные текстурные изображения

В качестве задачи обработки текстурных изображений рассмотрим задачу сегментации. Выберем размер тестовых изображений 201x201 пикселей. Формы областей текстур заранее заданы, и у нас есть данные о “правильной” сегментации. В качестве наблюдаемых признаков выберем красную и зеленую цветовые компоненты точек изображения. Признаки выбраны таким образом, чтобы можно было наглядно показать пространство наблюдаемых признаков с областями, соответствующими разным текстурам. Текстуры трех классов представляют собой реализации трех нормально распределенных случайных величин с одинаковыми дисперсиями и немного отличающимися средними в пространстве красной и зеленой цветовой компонент.

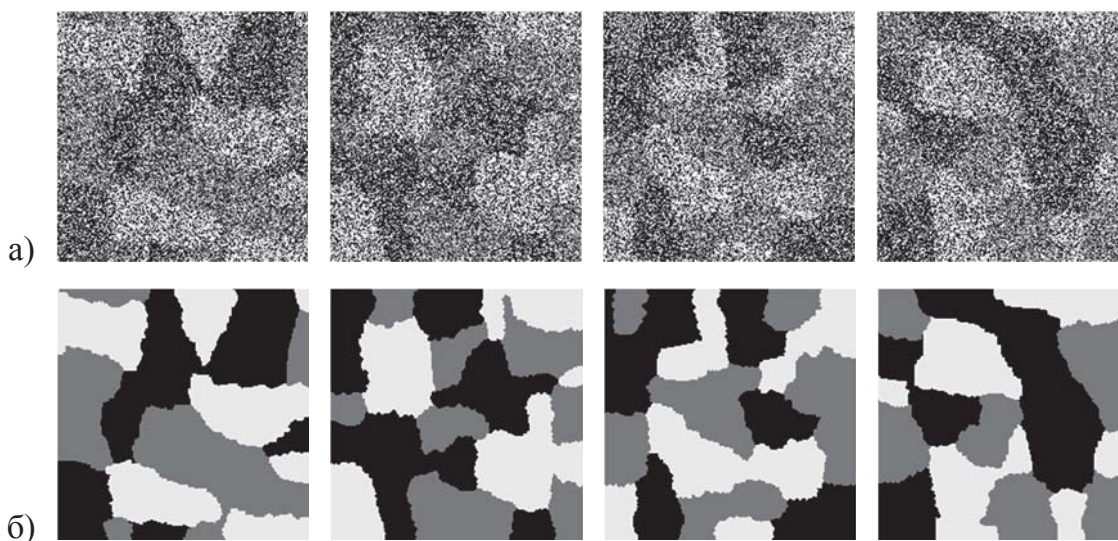


Рисунок 4.1 – Примеры тестовых изображений: а) текстурные изображения;  
б) классификация учителя

Взаимное расположение средних по множеству точек текстур каждого класса выбрано таким образом, что соответствующие области сильно пересекаются. На рис. 4.1 приведены примеры таких текстурных изображений.

Мы специально выбрали такие простые типы текстур, чтобы не ставить перед собой задачу выбора информативных признаков. Эта задача отвлекала

бы нас от основной проблемы – проблемы компенсации ошибок, связанных с древовидной аппроксимацией.

На рис. 4.2 приведено пространство наблюдаемых признаков для тестовых изображений.

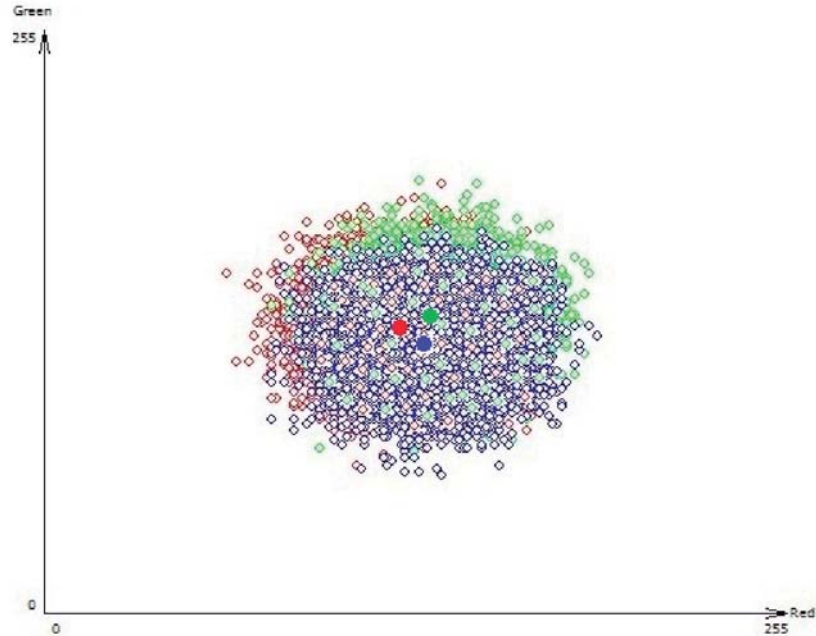


Рисунок 4.2 – Пространство наблюдаемых признаков

Как видно из рис. 4.2, в пространстве признаков области, соответствующие разным классам текстур, сильно пересекаются.

Задача независимого распознавания заключается в восстановлении параметров двумерного нормального распределения методом максимального правдоподобия. Многомерное нормальное распределение имеет вид:

$$f(\mathbf{y}) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{y}-\boldsymbol{\mu})}, \mathbf{y} \in \mathbf{R}^n, \quad (4.1)$$

где  $\Sigma(n \times n)$  – ковариационная матрица,  $\boldsymbol{\mu}(1 \times n)$  – вектор математического ожидания,  $n$  – размерность пространства [1].

Пусть обучающая совокупность содержит  $N$  объектов, которые имеют соответствующие векторы признаков  $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N$ , где  $\mathbf{y}_i = (y_{i1}, y_{i2}, \dots, y_{in})^T$ ,  $i = 1, \dots, N$ .

Параметры данного распределения могут быть восстановлены следующим образом:

$$\mu_i = \frac{1}{N} \sum_{j=1}^N y_{ji}, i = 1 \dots n,$$

$$\sigma_{ik} = \frac{1}{N} \sum_{j=1}^N (y_{ji} - \mu_i)(y_{jk} - \mu_k); i, k = 1, \dots, n. \quad (4.2)$$

Результаты независимого распознавания на таких текстурных изображениях достаточно плохие, где число ошибок составляет примерно 30%.

## 4.2 Сравнение алгоритмов подбора диагонального элемента для ациклического графа

Сравнивались все алгоритмы распознавания для пяти ациклических графов соседства (см. рис. 1.4). Графики найденных значений диагонального элемента и соответствующих чисел ошибок при обучении в случае ступенчатого дерева показаны на рис. 4.3–4.4.

Каждое тестовое изображение размером  $201 \times 201$  содержит 40401 пикселей. Тогда, например, число ошибок  $2000 \div 9000$  на одном изображении будет составлять  $5 \div 22\%$  от числа пикселей.

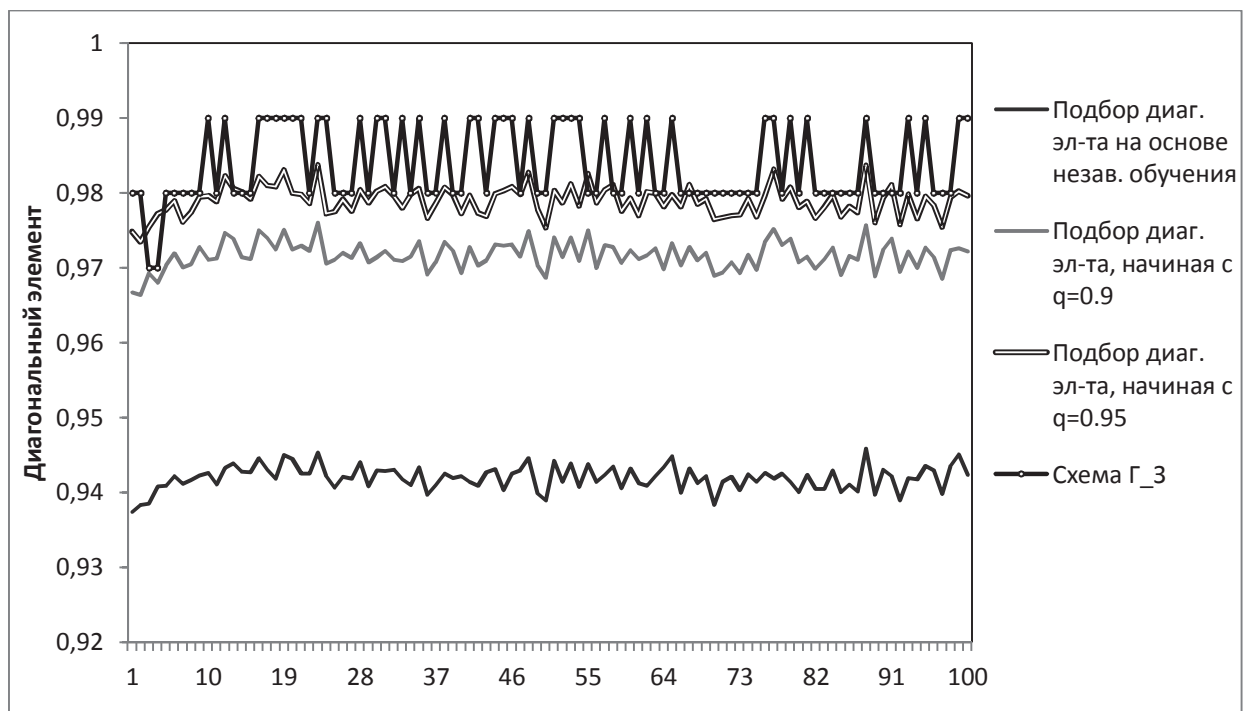


Рисунок 4.3 – Значения диагонального элемента в случае ступенчатого дерева

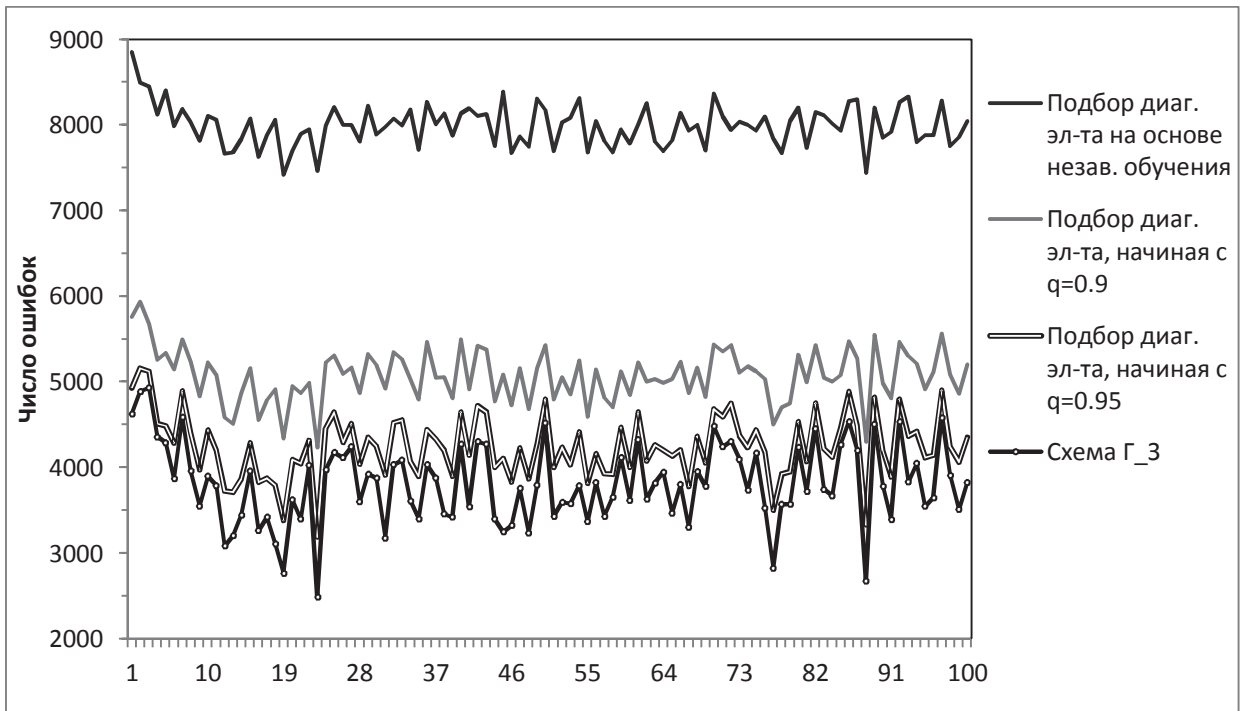


Рисунок 4.4 – Число ошибок при обучении в случае ступенчатого дерева

Графики найденных значений диагонального элемента и соответствующих чисел ошибок при обучении в случае спирали показаны на рис. 4.5–4.6.

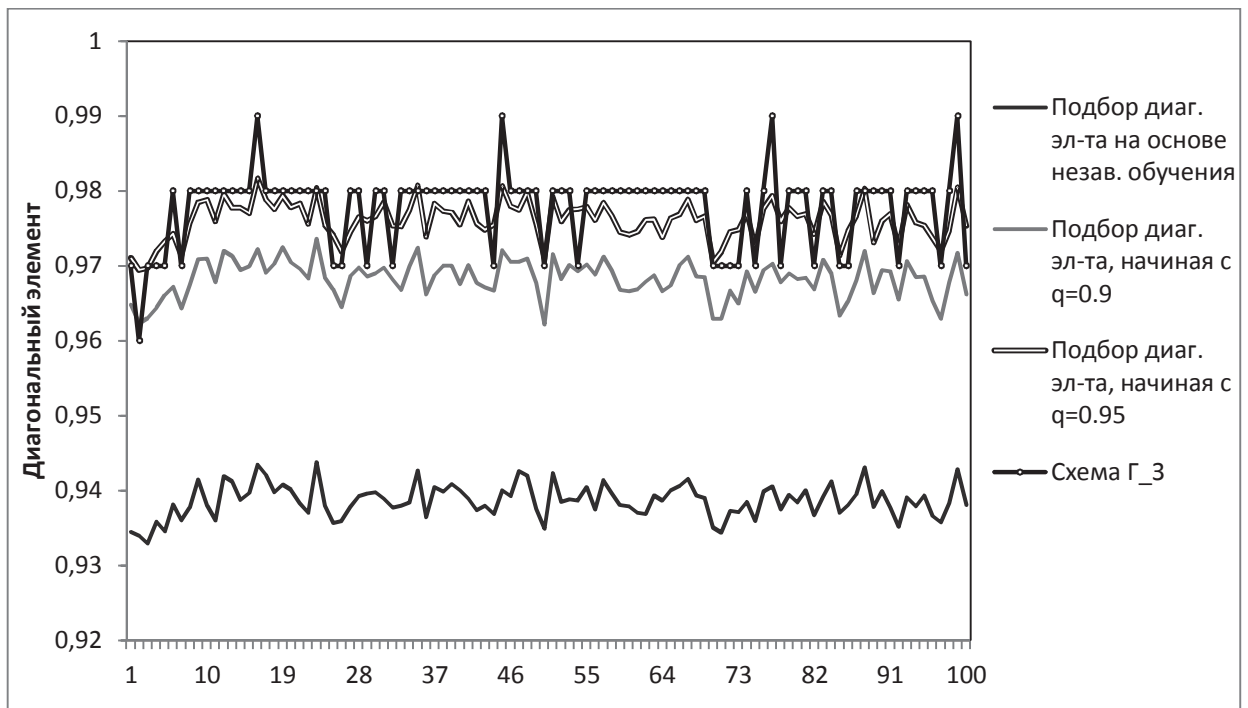


Рисунок 4.5 – Значения диагонального элемента в случае спирали

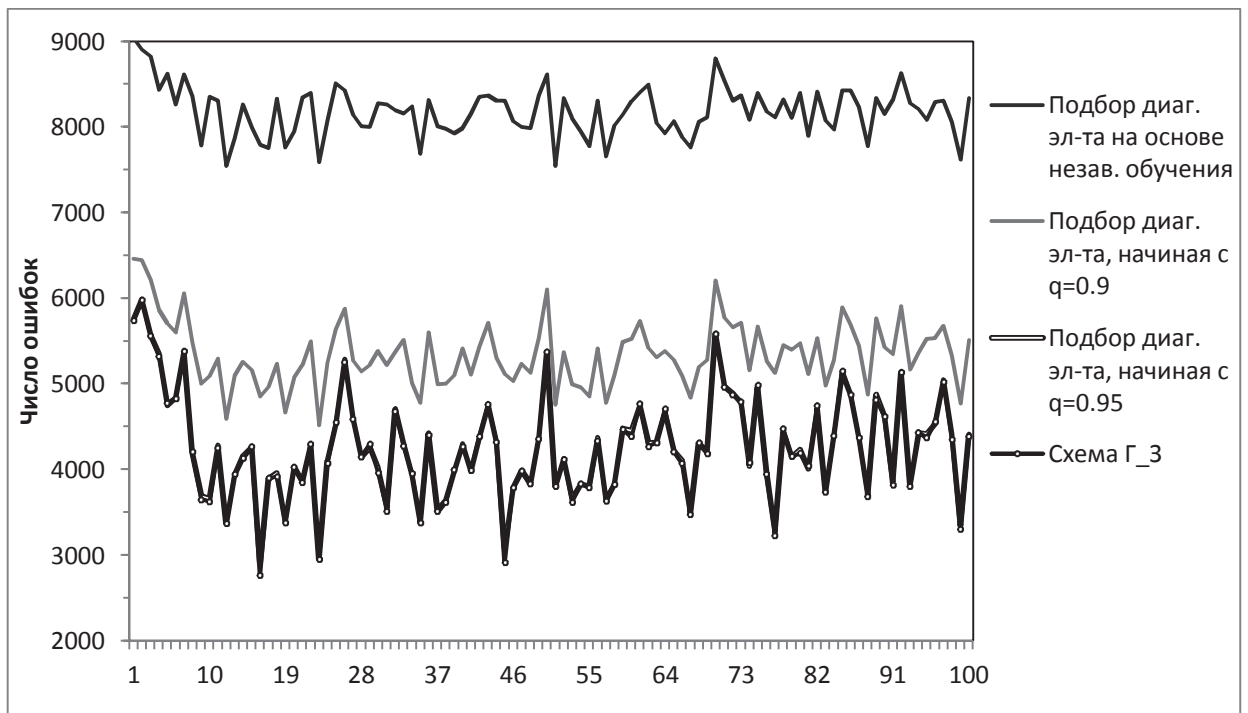


Рисунок 4.6 – Число ошибок при обучении в случае спирали

Графики найденных значений диагонального элемента и соответствующих чисел ошибок при обучении в случае горизонтальной змейки показаны на рис. 4.7–4.8.

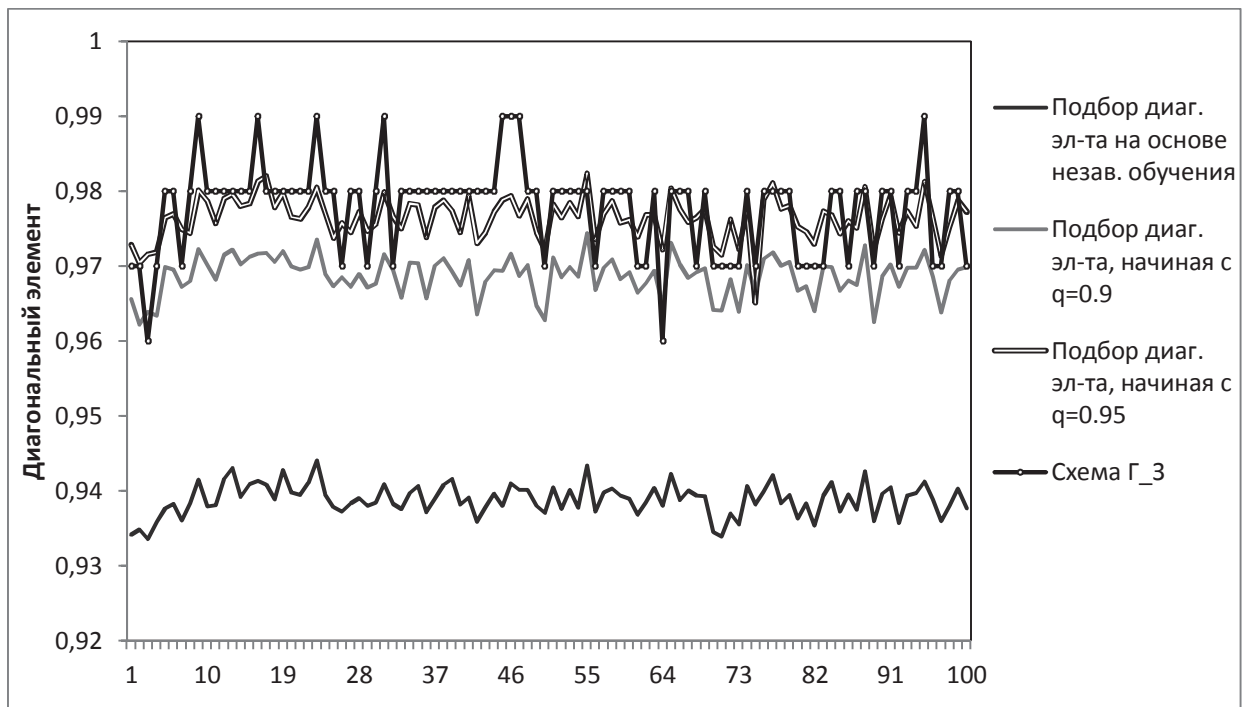


Рисунок 4.7 – Значения диагонального элемента в случае горизонтальной змейки

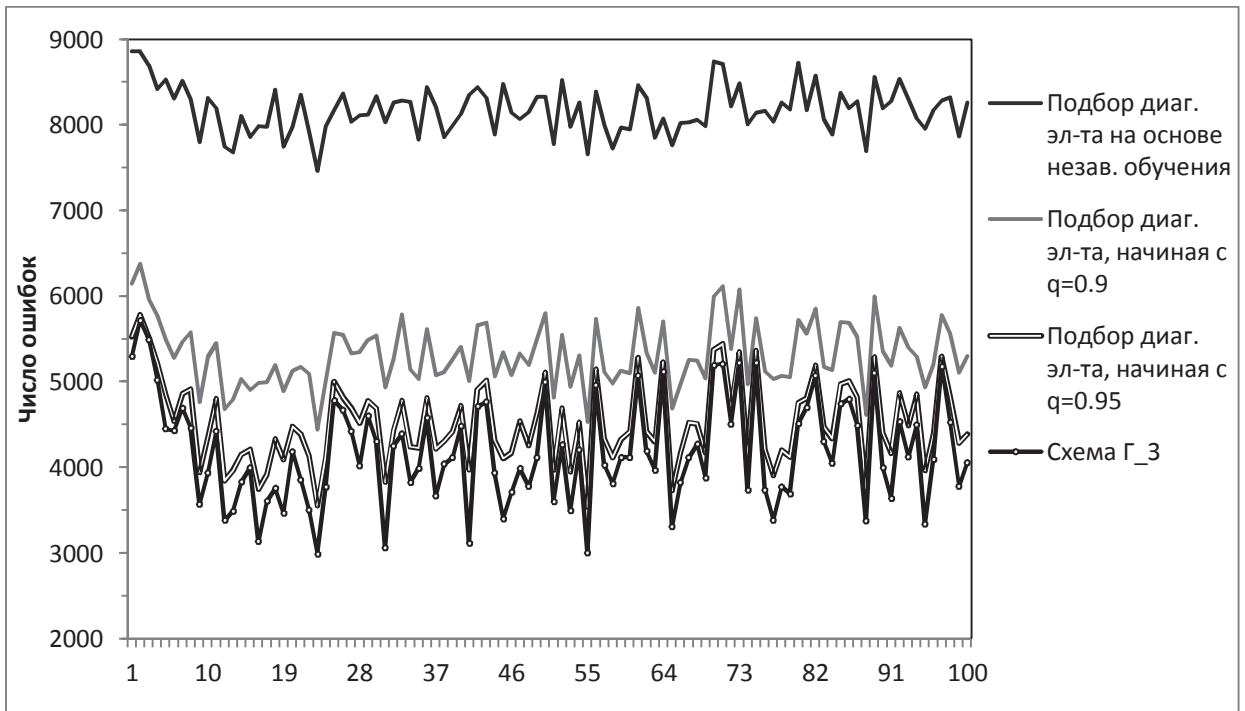


Рисунок 4.8 – Число ошибок при обучении в случае горизонтальной змейки

Графики найденных значений диагонального элемента и соответствующих чисел ошибок при обучении в случае вертикальной змейки показаны на рис. 4.9–4.10.

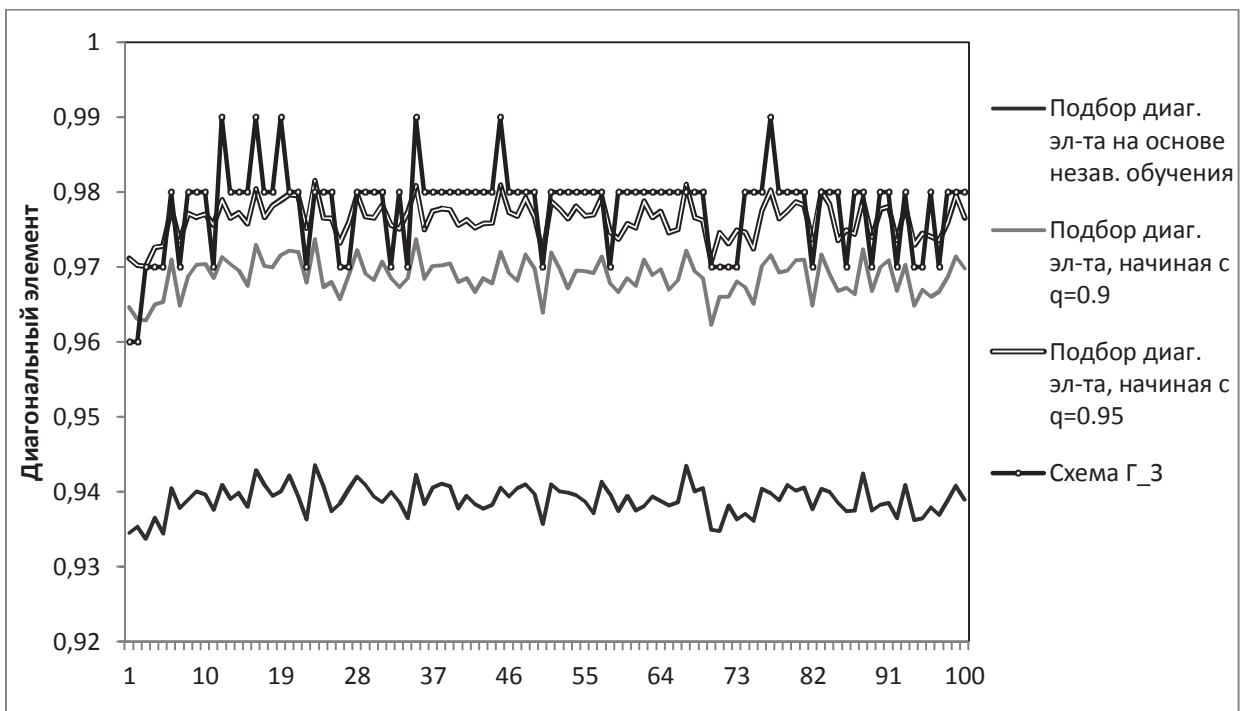


Рисунок 4.9 – Значения диагонального элемента в случае вертикальной змейки

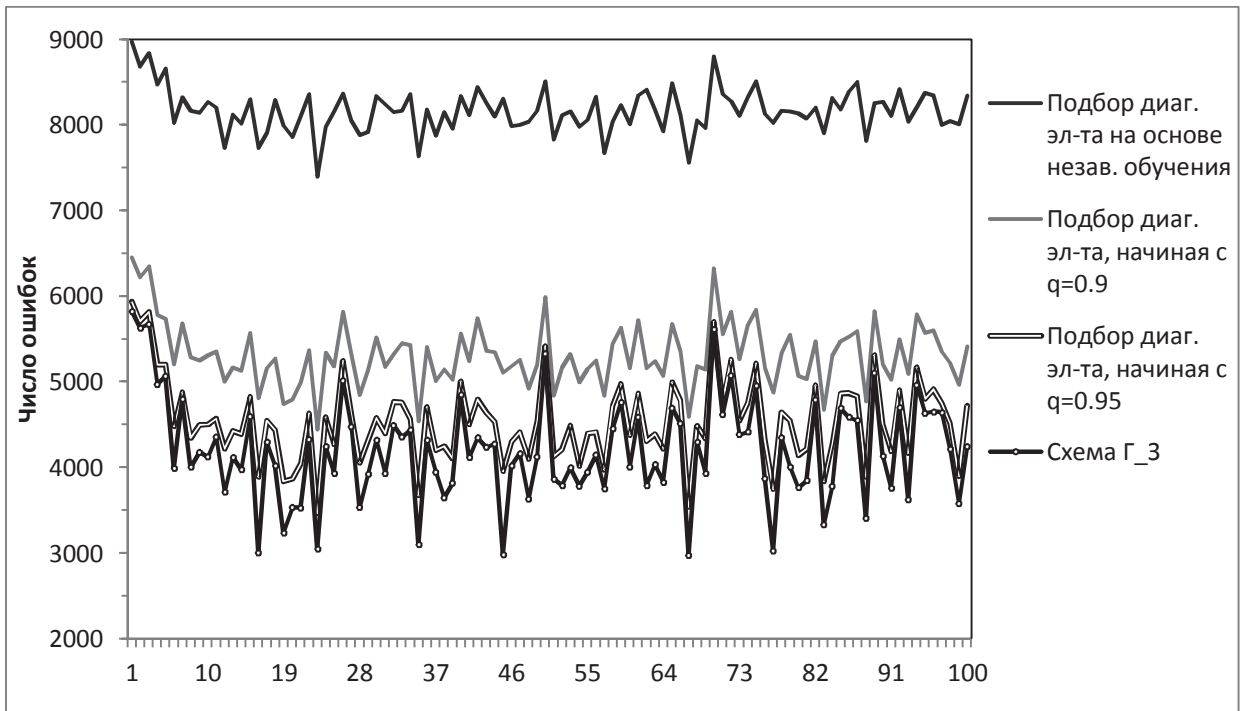


Рисунок 4.10 – Число ошибок при обучении в случае вертикальной змейки

Графики найденных значений диагонального элемента и соответствующих чисел ошибок при обучении в случае диагональной змейки показаны на рис. 4.11–4.12.

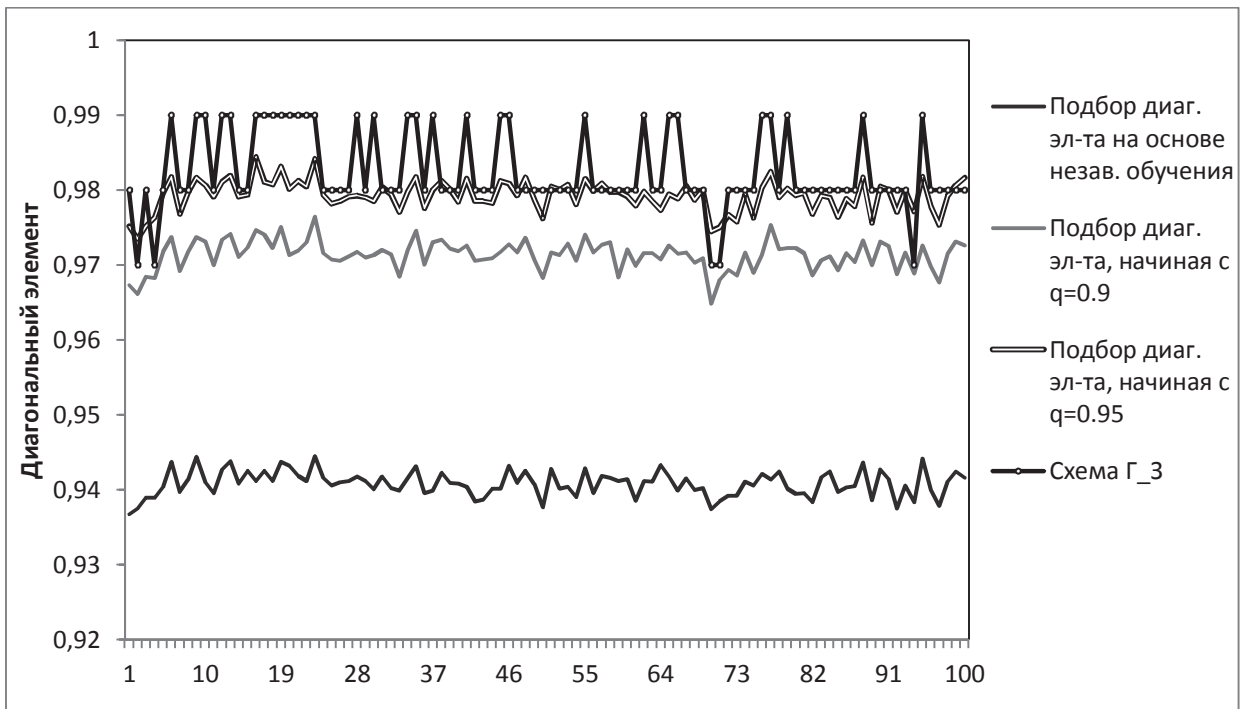


Рисунок 4.11 – Значения диагонального элемента в случае диагональной змейки

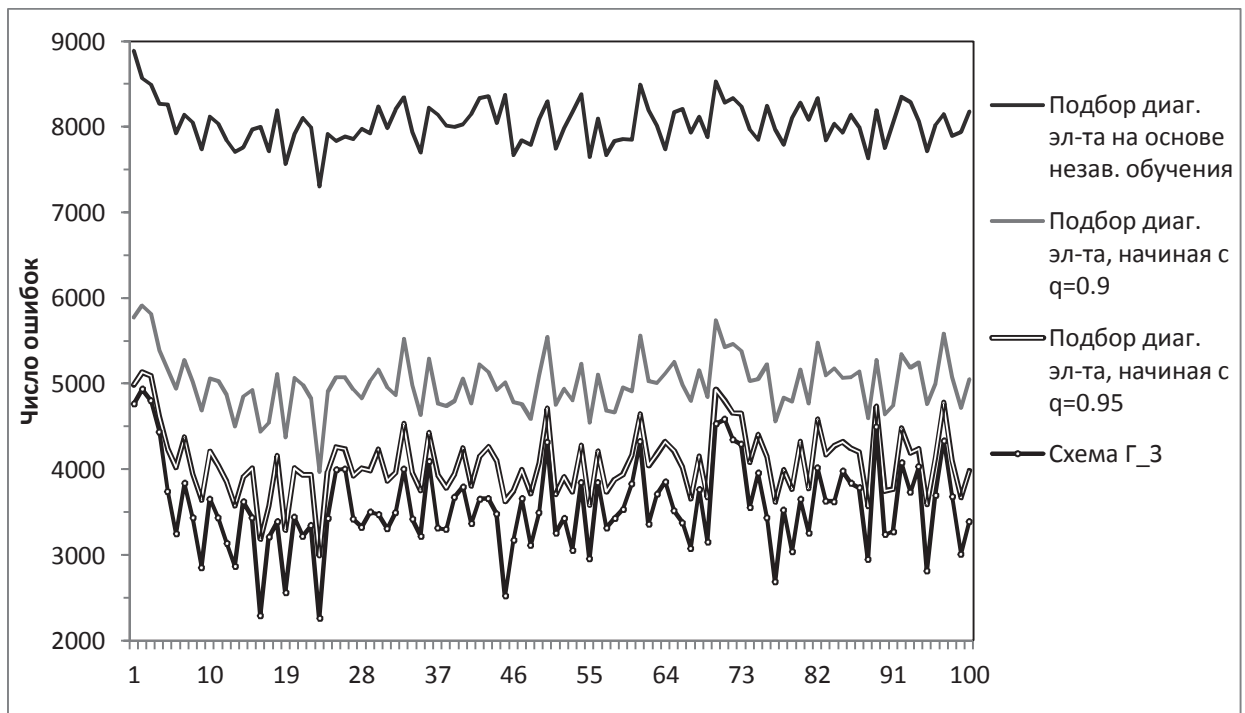


Рисунок 4.12 – Число ошибок при обучении в случае диагональной змейки

Из рис. 4.3–4.12 видно, что алгоритм подбора диагонального элемента на основе результата, полученного при независимом обучении, является наименее эффективным. Если в нем на второй итерации вместо пересчета результата независимого обучения для диагонального элемента использовать заранее заданное значение, то результат улучшается. Ошибка обучения в трех первых алгоритмах значительно выше, чем ошибка обучения по схеме Гаусса-Зайделя. Это означает, что итерационное решение задачи (2.2) и (2.8) не обеспечивает хорошей итоговой оценки ошибки по сравнению со схемой Гаусса-Зайделя. Причину этого мы объясним позже.

Тем не менее, главная цель заключается в подборе диагонального элемента. Имея подобранное значение диагонального элемента, следует применить базовый алгоритм для распознавания изображений и оценить ошибку. Результат сравнения среднего числа ошибок распознавания на 100 изображениях приведен на рис. 4.13–4.17.



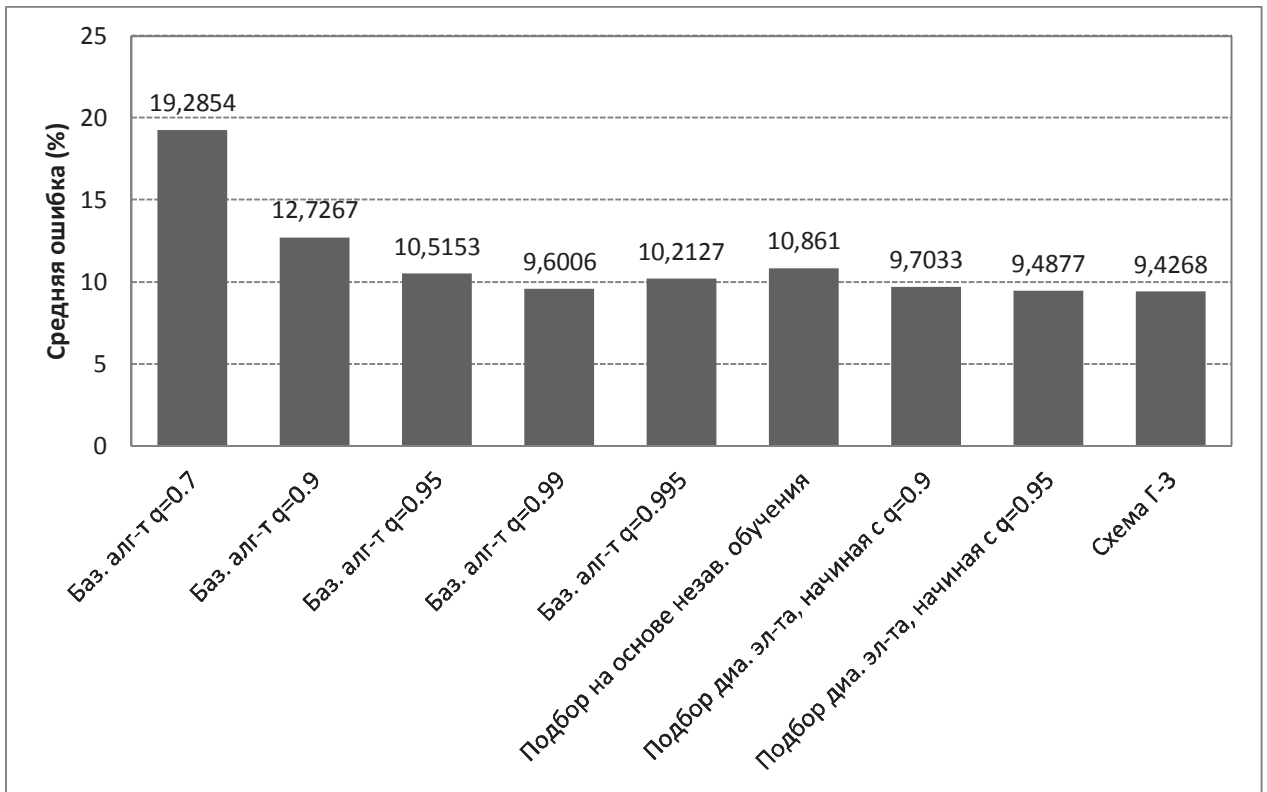


Рисунок 4.13 – Число ошибок распознавания в случае ступенчатого дера

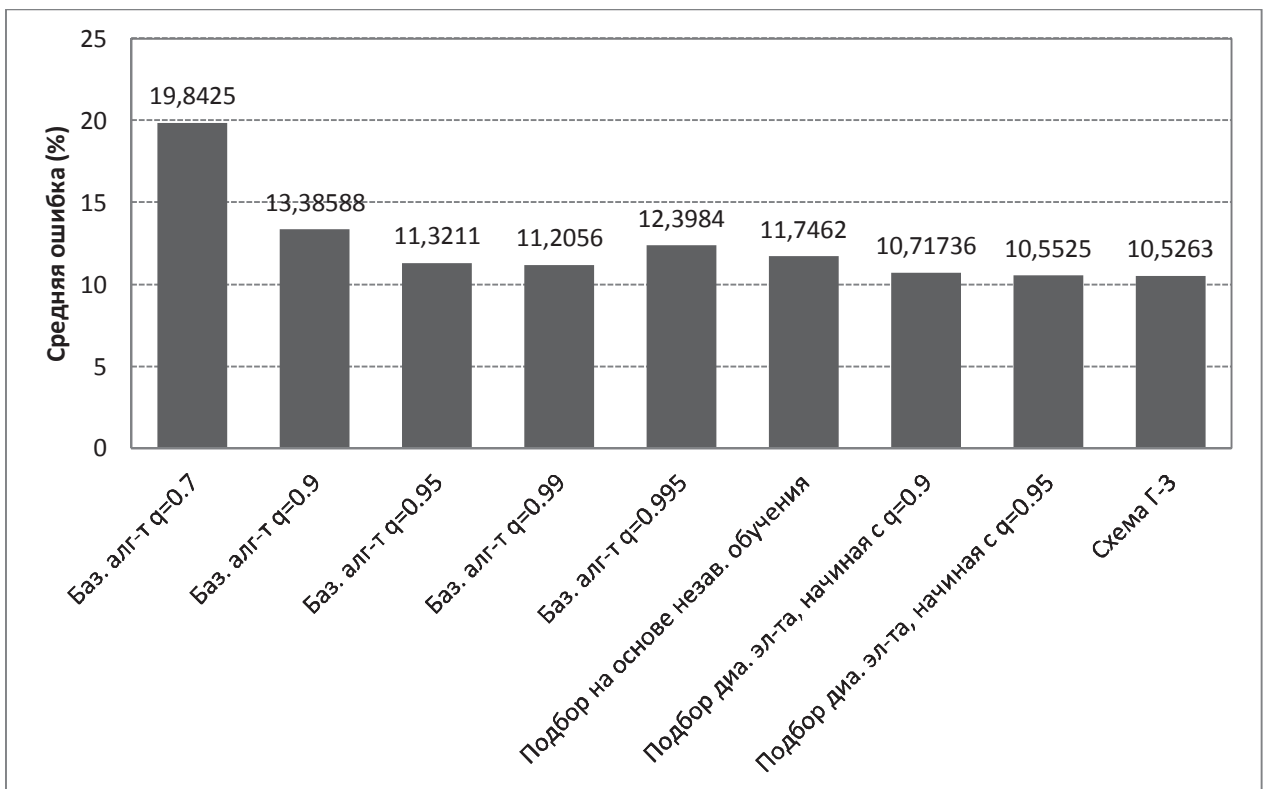


Рисунок 4.14 – Число ошибок распознавания в случае спирали

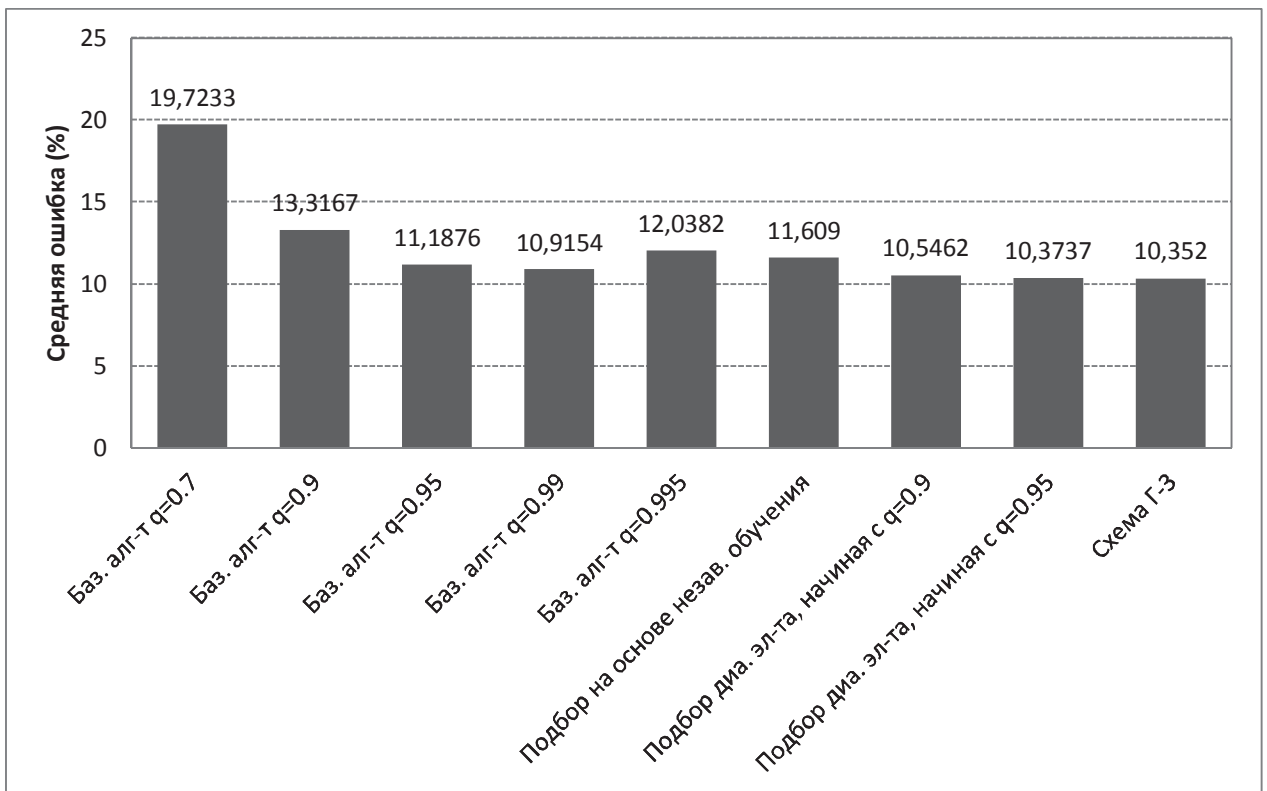


Рисунок 4.15 – Число ошибок распознавания в случае горизонтальной змейки

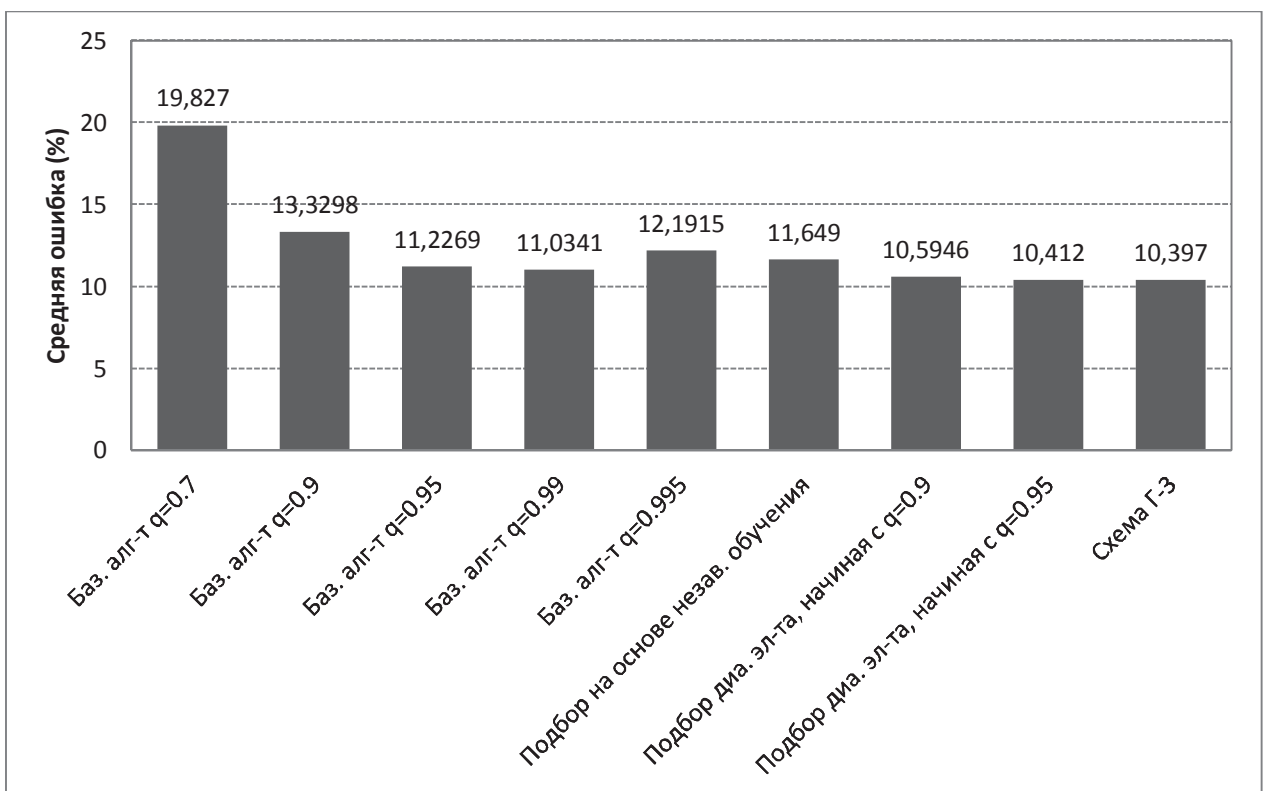


Рисунок 4.16 – Число ошибок распознавания в случае вертикальной змейки

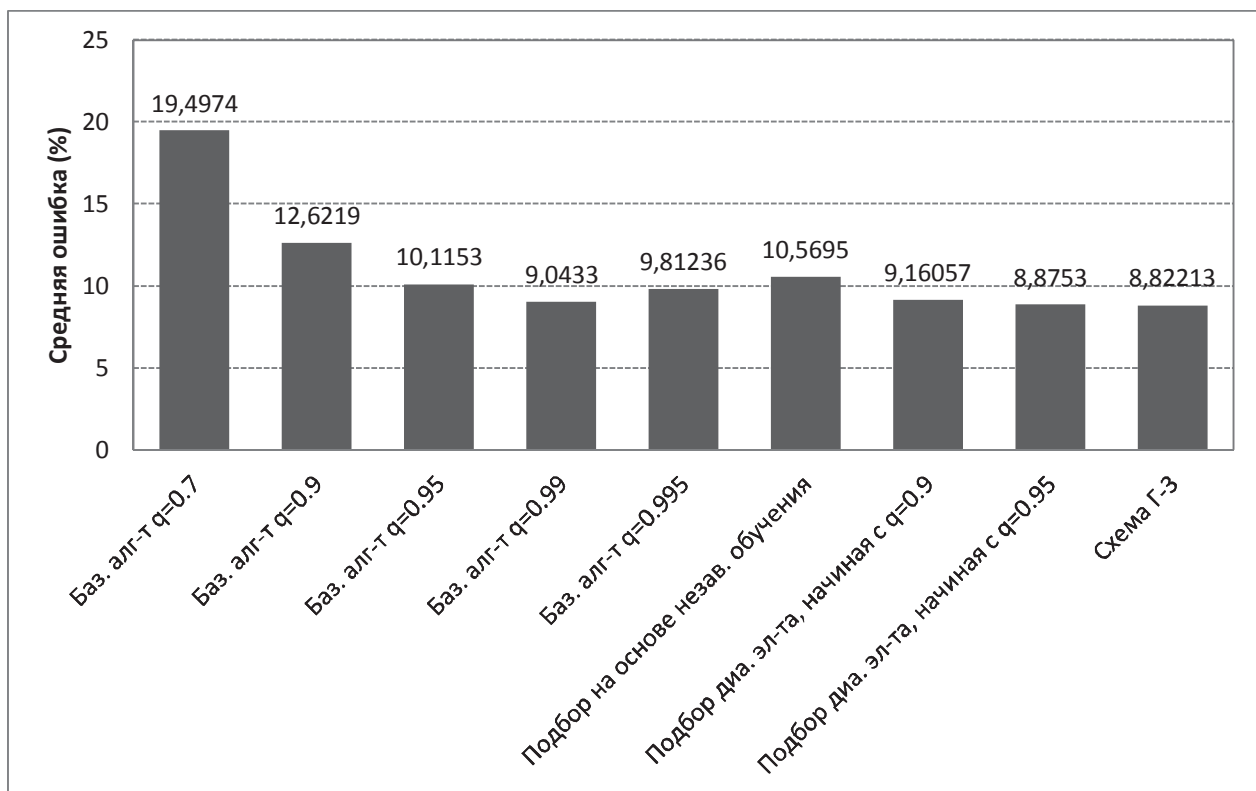


Рисунок 4.17 – Число ошибок распознавания в случае диагональной змейки

Из рис. 4.13–4.17 видно, что алгоритм подбора диагонального элемента на основе независимого обучения опять же является наихудшим. Если в нем на второй итерации вместо пересчитанного по независимому обучению диагонального элемента использовать заданный, то результат улучшается. Но есть случаи, когда алгоритм подбора диагонального элемента выигрывает у базового алгоритма и, наоборот, есть случаи, когда алгоритм подбора диагонального элемента проигрывает.

При повторении итераций базового алгоритма области каждого класса сливаются, а число пар соседних вершин с одинаковыми классами увеличивается, что увеличивает множество  $V_1$ . Поэтому при повторении итераций, согласно (2.9), фактически неявно пересчитывается диагональный элемент. Поэтому итерации позволяют улучшить качество распознавания.

В алгоритмах подбора диагонального элемента на основе (2.9) фактически дважды пересчитывается диагональный элемент: из-за слияния областей и непосредственно по (2.9). Это быстро увеличивает диагональный эле-

мент и приводит к быстрому слиянию областей. Можно предположить, что ускорение слияния областей отрицательно влияет на качество распознавания. Алгоритм по схеме Гаусса-Зайделя дает лучший результат и выигрывает у остальных алгоритмов. Оптимальное значение близко к единице (0.97–0.99).

### 4.3 Зависимость скорости сходимости от значения диагонального элемента

Результаты экспериментов показывают, что тем больше значение диагонального элемента, чем меньше необходимое число итераций базового алгоритма распознавания. Когда значение диагонального элемента достаточно близко к 1 (примерно более 0.97), то требуется лишь одна итерация базового алгоритма, чтобы получить окончательный результат распознавания.

В свою очередь, оптимальное значение диагонального элемента резко уменьшает число итераций (до одной – двух) базового алгоритма для любого ациклического графа соседства, так как оптимальное значение диагонального элемента обычно большое и близко к 1.

Например, для случая ступенчатого дерева график среднего числа итераций базового алгоритма на 10 первых изображениях приведен на рис. 4.18.

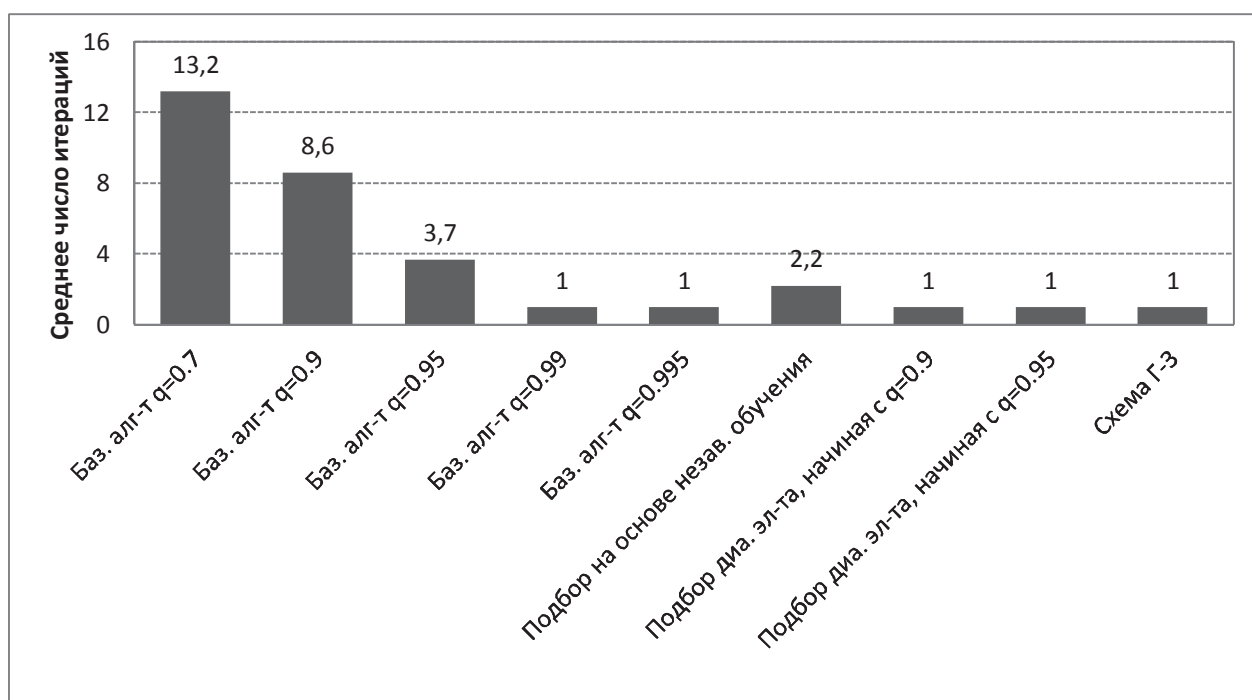


Рисунок 4.18 – Среднее число итераций в случае ступенчатого дерева

#### 4.4 Сравнение алгоритмов подбора параметров комбинирования ациклических графов соседства

Были проведены эксперименты на 100 модельных изображения для исследования свойств построенных алгоритмов подбора параметров комбинирования ациклических графов.

Для алгоритма подбора весов найденные веса графов показаны на рис. 4.19. Гистограмма средних значений подобранных весов графов и их СКО приведена на рис. 4.20.

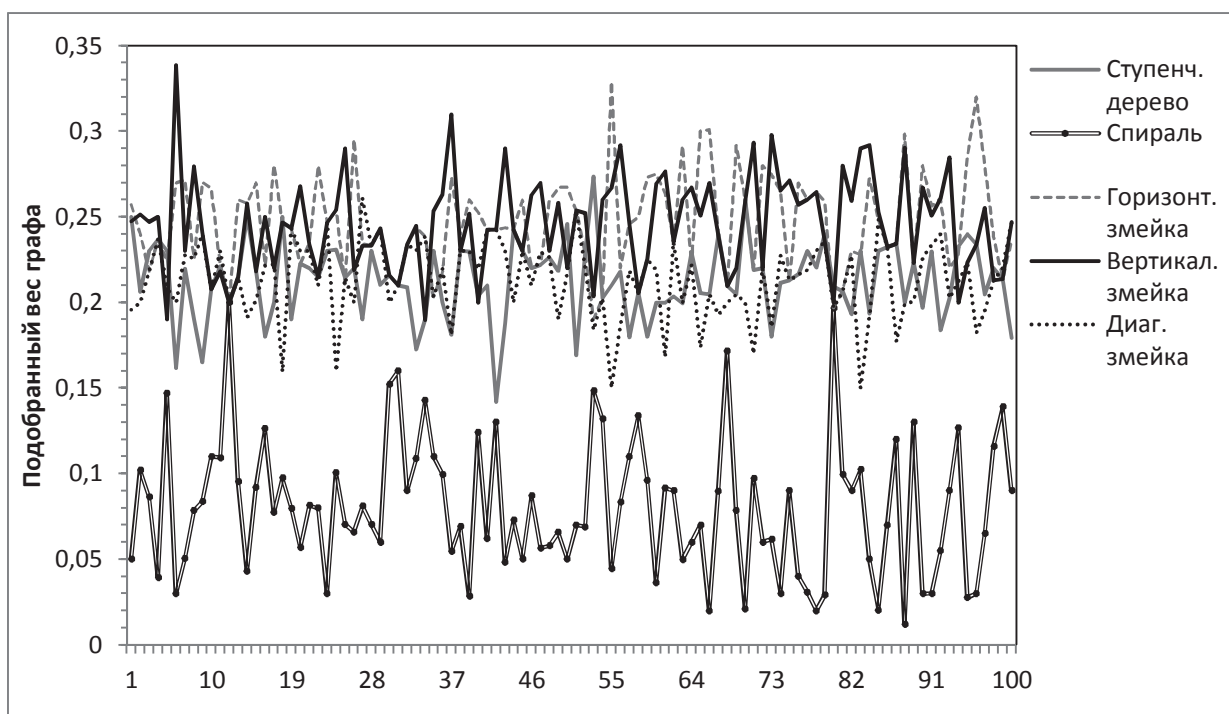


Рисунок 4.19 – Подобранные веса графов при алгоритме подбора весов с  $q = 0.95$

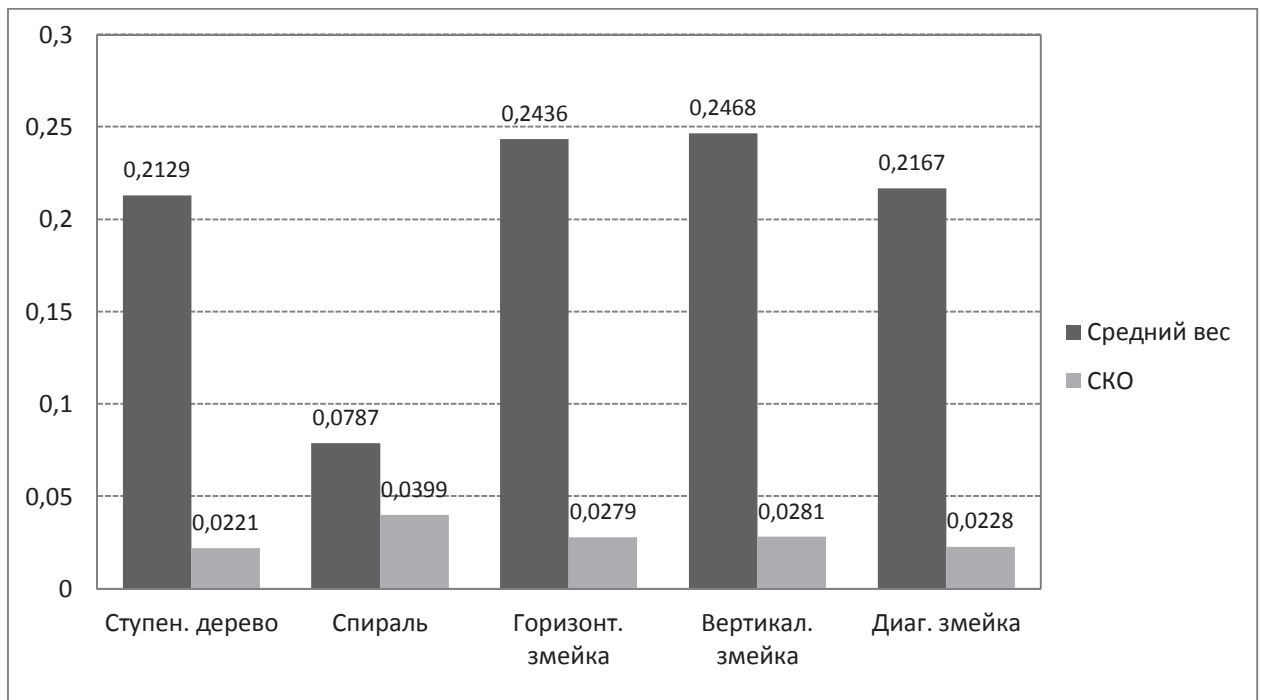


Рисунок 4.20 – Средние значения подобранных весов и их СКО при алгоритме подбора весов с  $q = 0.95$

Для алгоритма подбора параметров с единственным диагональным элементом, подобранные значения весов графов показаны на рис. 4.21. Гистограмма средних значений подобранных весов графов и их СКО приведена на рис. 4.22. Распределение оптимального значения диагонального элемента приведено на рис. 4.23.

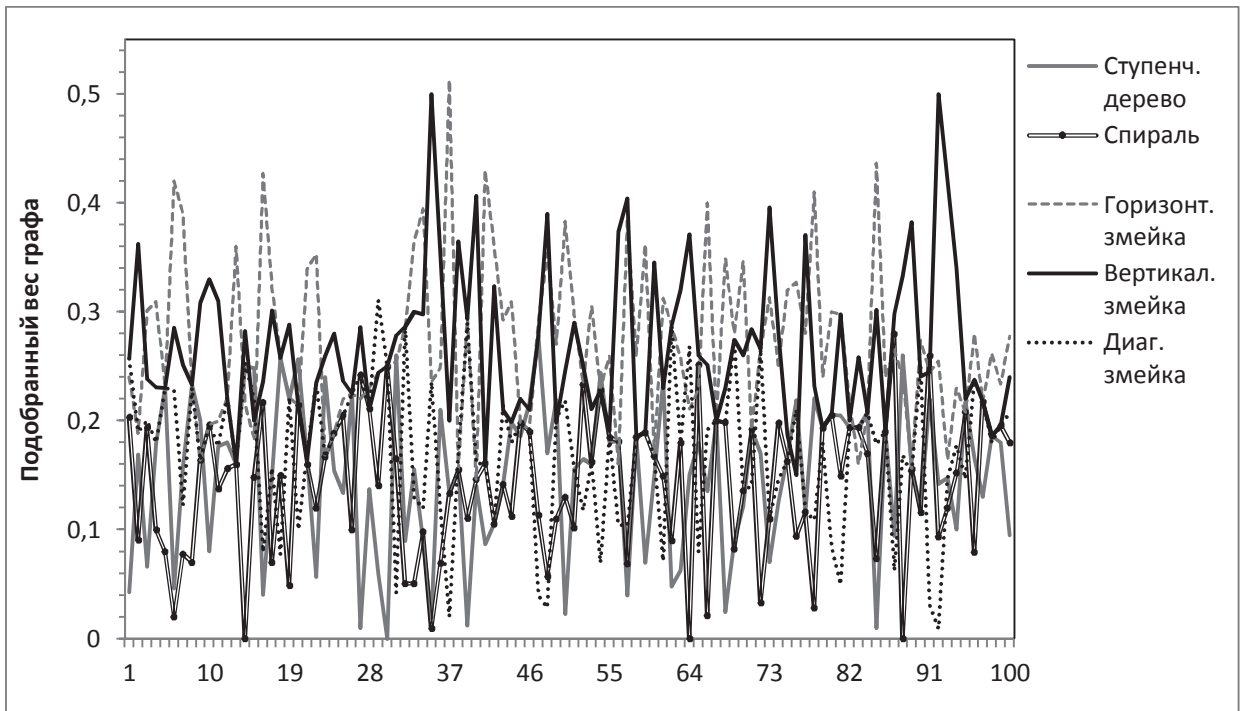


Рисунок 4.21 – Подобранные веса графов при алгоритме подбора параметров с единственным диагональным элементом

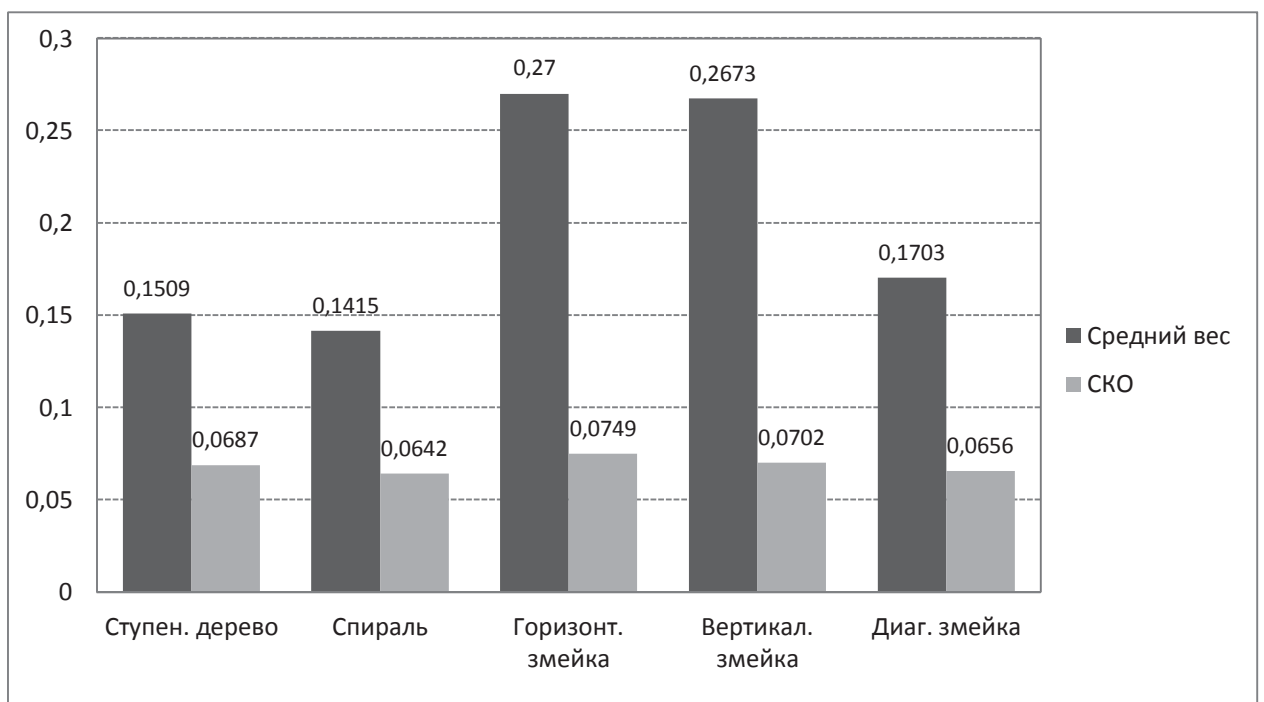


Рисунок 4.22 – Средние значения подобранных весов и их СКО при алгоритме подбора параметров с единственным диагональным элементом

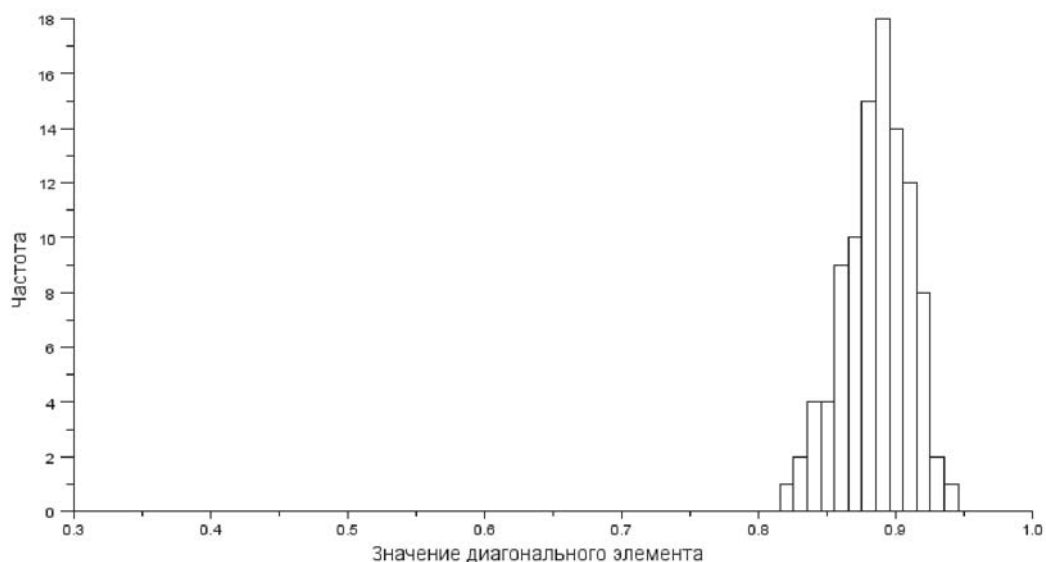


Рисунок 4.23 – Распределение значений оптимального диагонального элемента при алгоритме подбора параметров с единственным диагональным элементом

Для первой схемы подбора параметров с многими диагональными элементами, подобранные значения весов графов показаны на рис. 4.24. Гистограмма средних значений подобранных весов графов и их СКО приведена на рис. 4.25. Распределение оптимального значения диагональных элементов приведено на рис. 4.26.

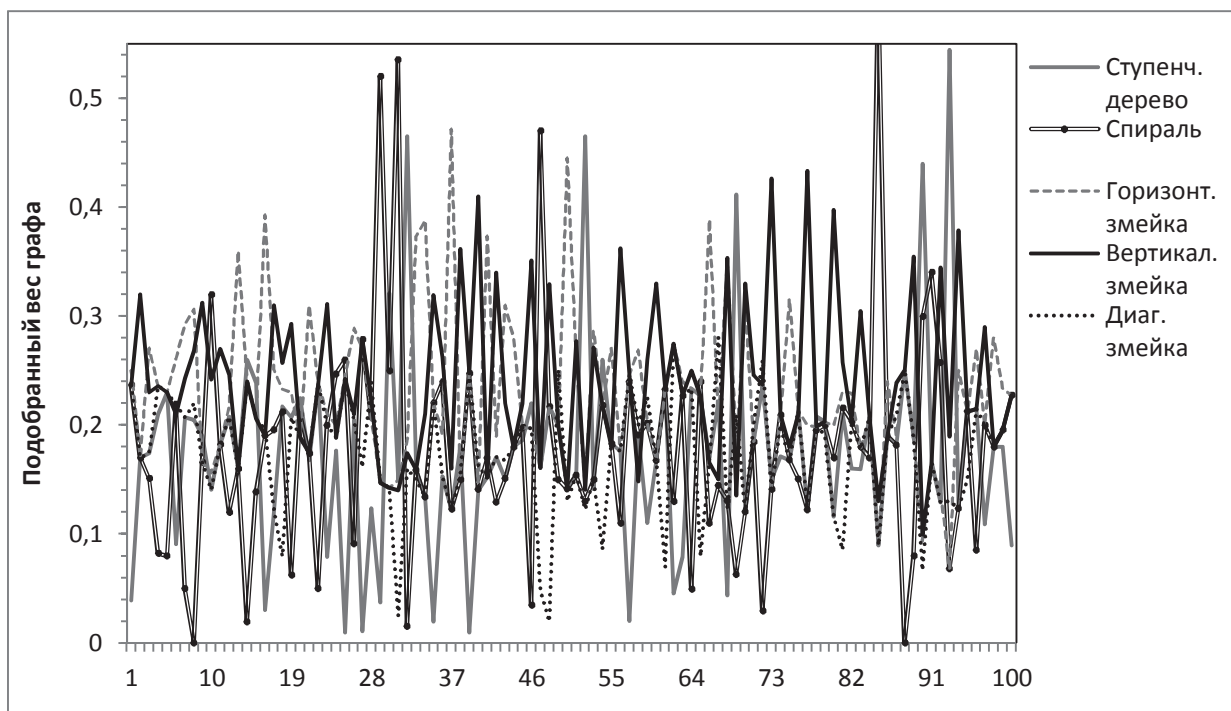


Рисунок 4.24 – Подобранные веса графов при первой схеме подбора параметров со многими диагональными элементами



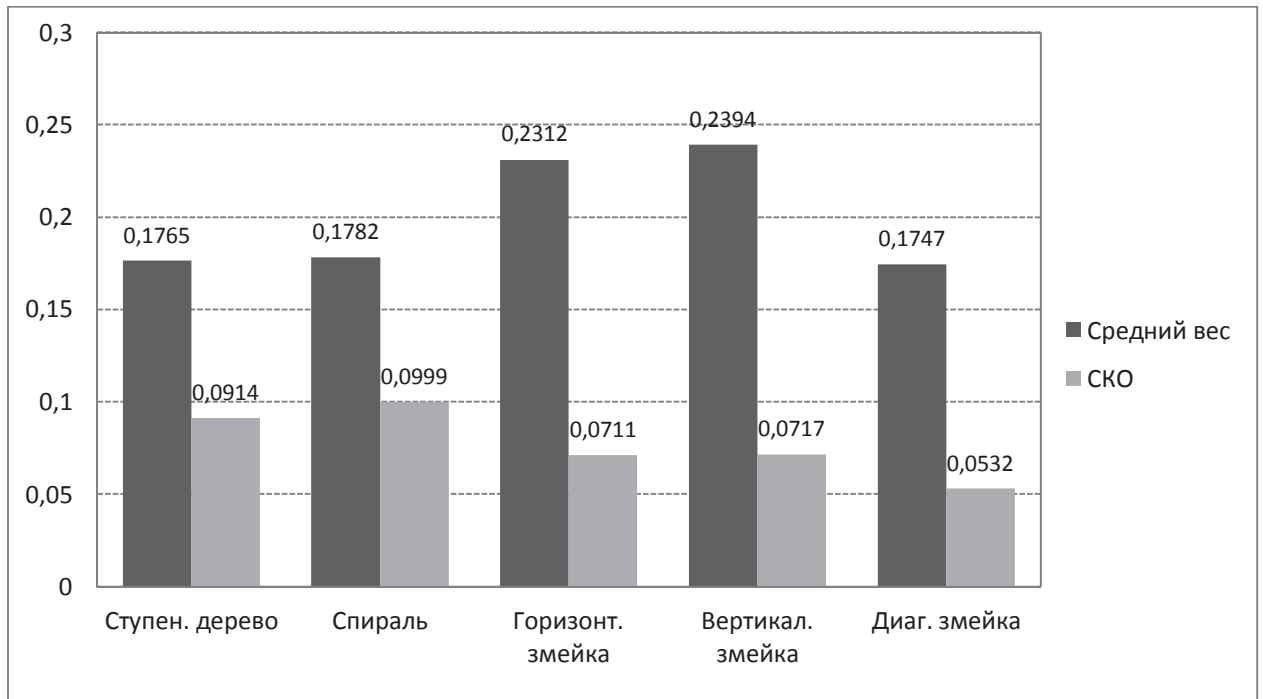


Рисунок 4.25 – Средние значения подобранных весов и их СКО при первой схеме подбора параметров со многими диагональными элементами

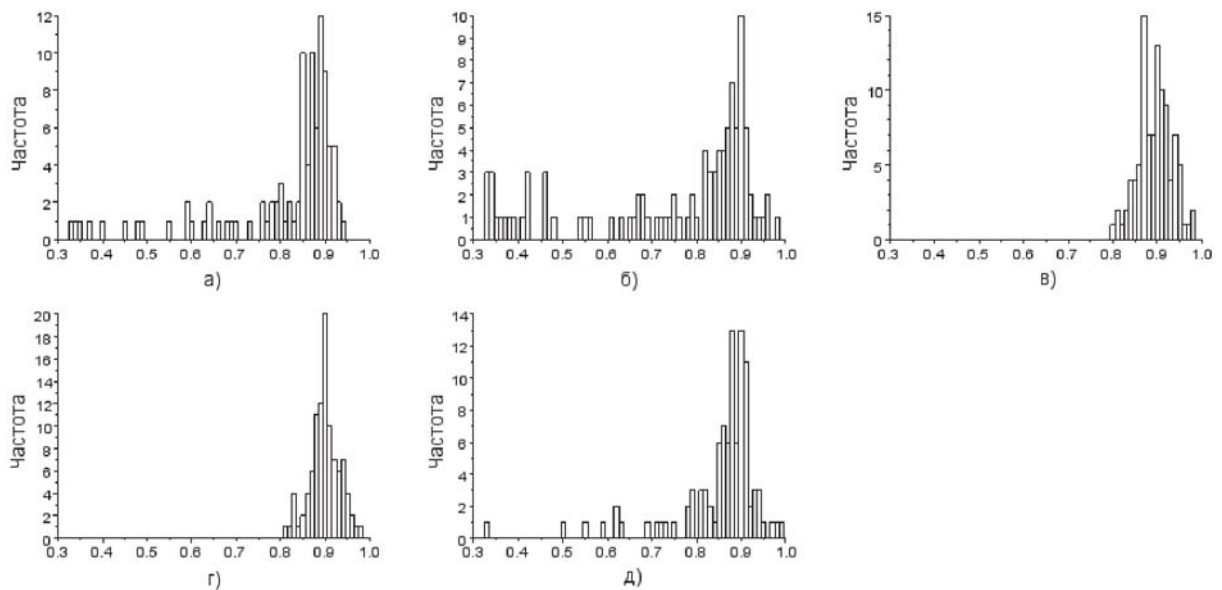


Рисунок 4.26 – Распределение оптимальных значений диагональных элементов при первой схеме подбора параметров со многими диагональными элементами: а) для ступенчатого дерева; б) для спирали; в) для горизонтальной змейки; г) для вертикальной змейки; д) для диагональной змейки

Для второй схемы подбора параметров с многими диагональными элементами, подобранные значения весов графов показаны на рис. 4.27. Гисто-

грамма средних значений подобранных весов графов и их СКО приведена на рис. 4.28. Распределение оптимального значения диагональных элементов приведено на рис. 4.29.

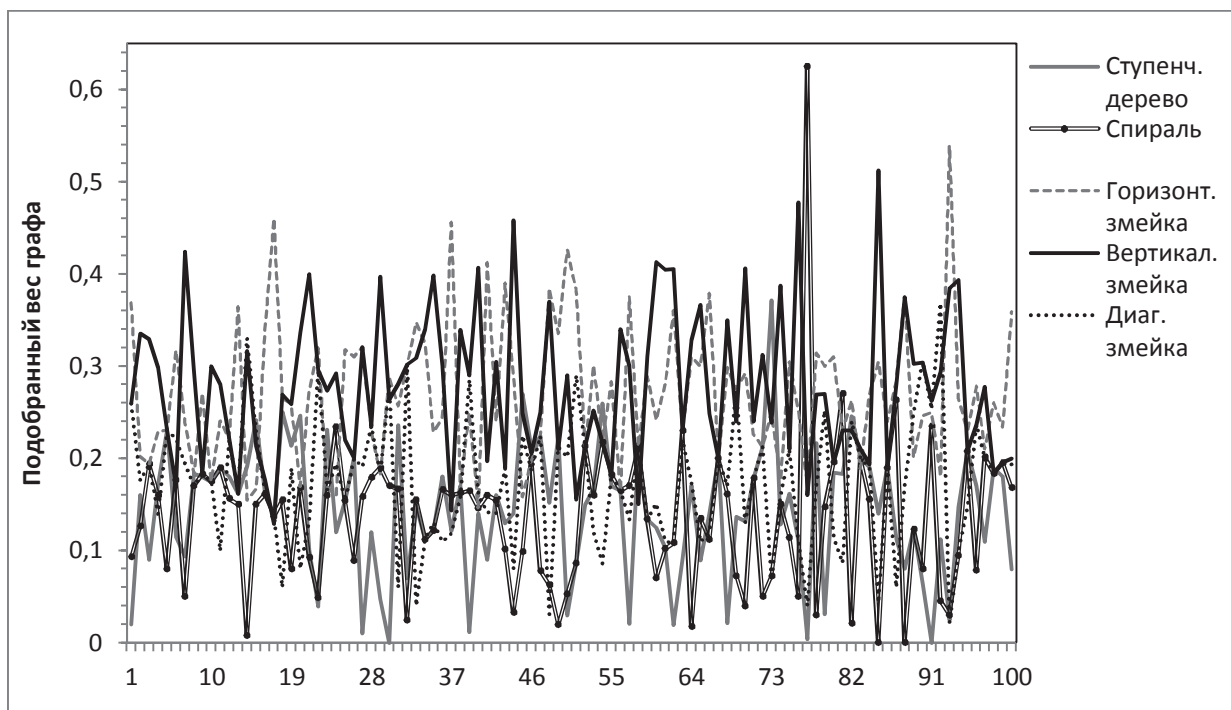


Рисунок 4.27 – Подобранные веса графов при второй схеме подбора параметров со многими диагональными элементами

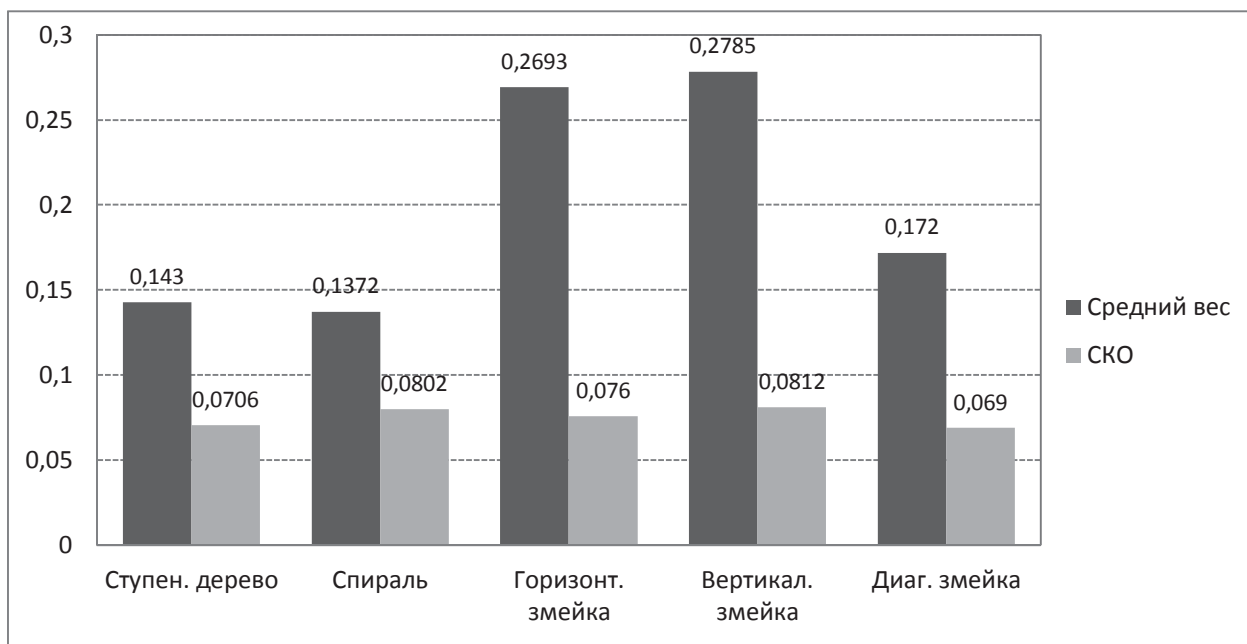


Рисунок 4.28 – Средние значения подобранных весов и их СКО при второй схеме подбора параметров со многими диагональными элементами

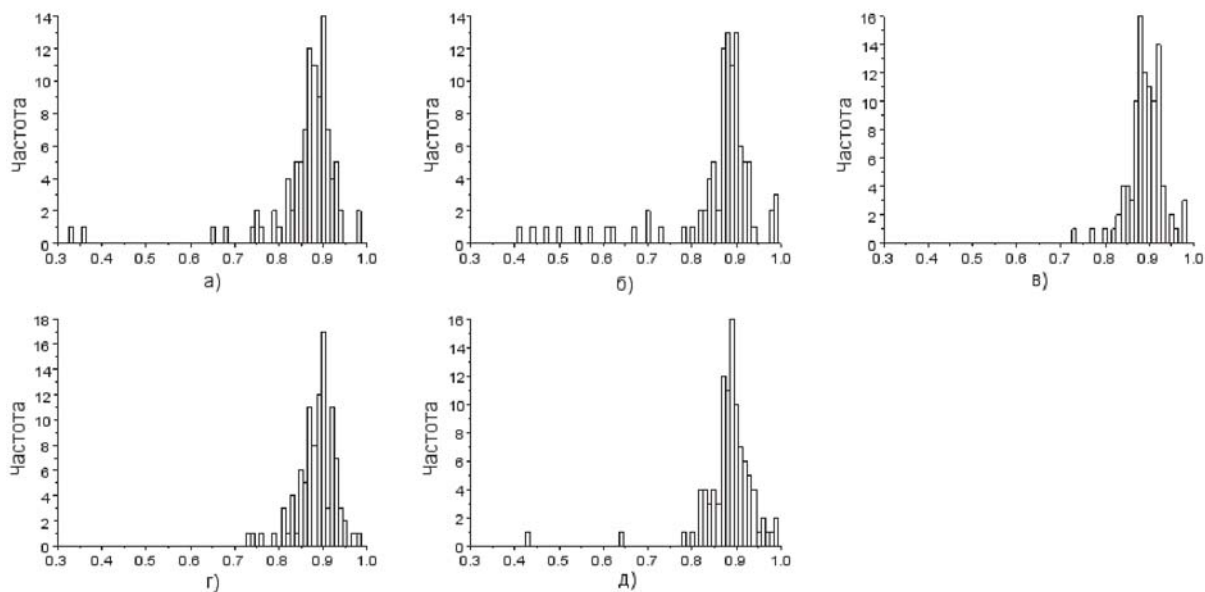


Рисунок 4.29 – Распределение оптимальных значений диагональных элементов при второй схеме подбора параметров со многими диагональными элементами: а) для ступенчатого дерева; б) для спирали; в) для горизонтальной змейки; г) для вертикальной змейки; д) для диагональной змейки

Из рис. 4.19–4.20 видно, что при подборе весов с ранее заданным значением диагонального элемента  $q = 0.95$  средний вес ( $w_2 = 0.079$ ) спирали является наименьшим среди весов всех графов. Все остальные имеют примерно одинаковые средние веса. Причем средние веса ( $w_3, w_4 \approx 0.245$ ) горизонтальной и вертикальной змеек чуть выше, чем средние веса ( $w_1, w_5 \approx 0.215$ ) ступенчатого дерева и диагональной змейки. Стандартное отклонение (СКО) оптимального веса у каждого графа достаточно мало ( $\delta_w = 0.02 - 0.04$ ). Это означает, что разница между подобранными весами на разных изображениях небольшая.

С одной стороны, раньше в разделе 3.1 данной работы, согласно результатам предварительных экспериментов, было показано, что при подборе весов, в котором используется однократное распознавания, при изменении значения диагонального элемента наименьшее число ошибок достигается фактически при одном и том же весе графа. Это значит, что подобранные значения параметров фактически не изменяются при варьировании диаго-

нального элемента матрицы условных вероятностей переходов. Можно понимать стабилизацию подобранных значений параметров у алгоритма подбора весов как «устойчивость» работы алгоритма.

С другой стороны, результаты распознавания на 100 изображениях показывают, что подобранные значения параметров на разных изображениях также слабо меняются (СКО выбранного веса у каждого графа достаточно мало  $\delta_w = 0.02 - 0.04$ ). Этот факт позволяет снова утвердить «устойчивость» работы алгоритма подбора весов не только при разных значениях диагонального элемента, но и на разных изображениях. Хотя среди всех сравнимых алгоритмов, результат подбора весов является наихудшим.

Из рис. 4.21–4.23 видно, что при добавлении одного параметра (единственного диагонального элемента) средние веса горизонтальной и вертикальной змеек чуть повышается ( $w_3, w_4 \approx 0.27$ ), средний вес спирали также увеличивается ( $w_2 \approx 0.14$ ), в средние веса дерева и диагональной змейки уменьшаются ( $w_1, w_5 \approx 0.16$ ). Величина СКО выбранного веса каждого графа также увеличивается, т.е. диапазон оптимальных весов расширяется. Среднее значение единственного диагонального элемента  $q = 0.8858$  и его СКО достаточно мало ( $\delta_q = 0.0245$ ). Оптимальное значение диагонального элемента находится в диапазоне  $[0.83; 0.93]$ .

Из рис. 4.24–4.29 видно, что в двух схемах подбора параметров со многими диагональными элементами средние веса графов остаются такими же, что и в алгоритме подбора с единственным элементом. СКО подобранных весов в этих схемах стали еще больше, т.е. подобранные веса меняются в более широких диапазонах. Средние значения диагональных элементов остаются примерно такими же, но их СКО также увеличиваются по сравнению с алгоритмом подбора с одним диагональным элементом.

Кроме этого, важно подчеркнуть, что в результатах двух схем подбора со многими элементами возникают «странные» оптимальные значения диагональных элементов, которые принадлежат диапазону  $q \in [0.33; 0.5]$ . Называ-

ем их «странными», так как мы знаем, что малое значение диагонального элемента не дает хорошего результата распознавания, и поэтому естественным образом мы ожидаем достаточно большие подобранные значения диагональных элементов.

Таким образом, по критерию о постоянстве подобранных значений параметров алгоритм подбора весов является самым устойчивым. Алгоритм подбора с единственным элементом менее устойчивый, и две схемы подбора со многими элементами являются самыми неустойчивыми.

Разницу степеней «устойчивости» этих алгоритмов можно объяснить по следующим соображениям. Функция числа ошибок  $E(\mathbf{w}, \mathbf{q})$ , зависящая от весов графов  $\mathbf{w}$  и диагональных элементов  $\mathbf{q}$ , является многомерной и многоэкстремальной. А ее область значений является дискретной (т.к. число ошибок должно быть натуральным). Поэтому мощность множества определения (число возможных комбинаций параметров) сравнительно во много раз больше, чем мощность множества значений (число возможных значений числа ошибок). Это значит, что для каждого значения числа ошибок могут существовать многие комбинации параметров, которые соответствуют данному значению числа ошибок. В результате чего существуют многие локальные минимальные точки (комбинации параметров), которые обладают одним и тем же уровнем ошибок (т.е. имеют одинаковое число ошибок).

При расширении числа параметров (добавлении диагональных элементов), мощность множества определения в экспоненциальное число раз увеличивается, а множество значений остается таким же. Поэтому число локальных минимальных точек так же резко увеличивается. В результате чего число локальных минимальных точек с одинаковым уровнем ошибки тоже увеличивается. Причем возникают и экстремальные комбинации параметров, в которых диагональные элементы имеют малые значения. Предложенные алгоритмы подбора параметров построены по идее покоординатного спуска Гаусса-Зайделя. Поэтому в зависимости от стратегии варьирования они могут перескакивать с одной локальной минимальной области на другую, и в итоге,

попадать в минимальную область с малыми значениями диагональных элементов.

Более того, в наших алгоритмах, при варьировании и нахождении значения диагонального элемента (или веса), которое обеспечивает наименьшее число ошибок, мы применяем следующую стратегию. Если существуют разные значения диагонального элемента (или веса), при которых достигается наименьшее число ошибок, то среди них мы берем наименьшее значение. Поэтому вероятность попадания в локальную точку с малыми значениями диагональных элементов становится больше при такой стратегии.

Тем не менее, по критерию о качестве распознавания результаты экспериментов на рис. 4.30–4.35 показывают другую ситуацию.

Представим экспериментальные результаты в виде зависимости среднего числа ошибок от числа распознаваемых изображений. Причем один алгоритм соответствует одной такой линии зависимости.

Пусть изображения индексированы с 1 до 100, а  $(s_1, s_2, \dots, s_{100})$  – это случайная перестановка последовательности индексов изображений  $(1, 2, \dots, 100)$ . Тогда данная зависимость представлена как линия, соединяющая последовательные точки, где  $k$ -ая точка – среднее число ошибок на  $k$  изображениях с индексами  $s_1, s_2, \dots, s_k$  ( $k=1, \dots, 100$ ). То есть:

1-ая точка – число ошибок на  $s_1$ -ом изображении;

2-ая точка – среднее число ошибок на  $s_1$ -ом и  $s_2$ -ом изображениях;

3-я точка – среднее число ошибок на  $s_1$ -ом,  $s_2$ -ом,  $s_3$ -ом изображениях;

.....

$k$ -ая точка – среднее число ошибок на  $s_1$ -ом,  $s_2$ -ом, ...,  $s_k$ -ом изображениях;

.....

100-ая точка – среднее число ошибок на всех 100 изображениях.

**Первый случай:**  $(s_1, s_2, \dots, s_{100})$  представляет собой обычный порядок индексов изображений, т.е.  $(s_1, s_2, \dots, s_{100}) = (1, 2, \dots, 100)$ .

Тогда соответствующий график таких линий средних чисел ошибок (в процентах) приведен ниже (рис. 4.30).

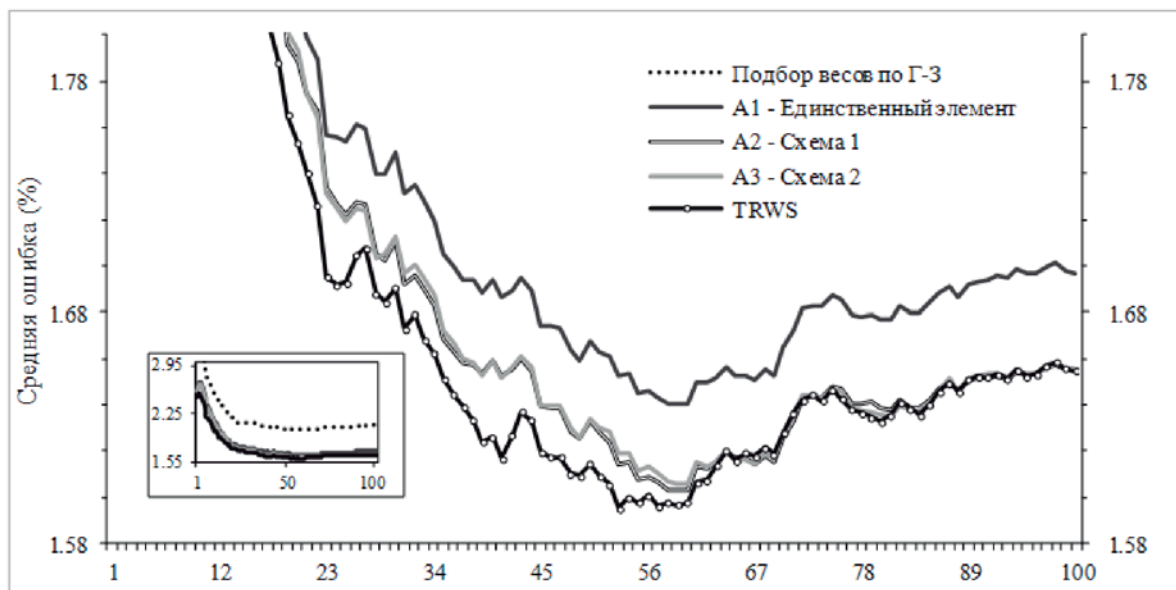


Рисунок 4.30 – График линий средних чисел ошибок при обычном порядке просмотра изображений

Из рис. 4.30 легко увидеть одинаковый характер линий ошибок для всех алгоритмов. Алгоритмы с подбором диагонального элемента  $q$  резко улучшают качество распознавания по сравнению с исходным без подбора. Алгоритм TRWS решает задачу лучше. Но начиная с каких-то изображений, он решает ее так же, т.е. наши алгоритмы решают задачу не хуже, особенно первая и вторая схемы со многими элементами. Это так же сразу видно из графика отклонения средней ошибки от алгоритма TRWS, показанного на рис. 4.31.

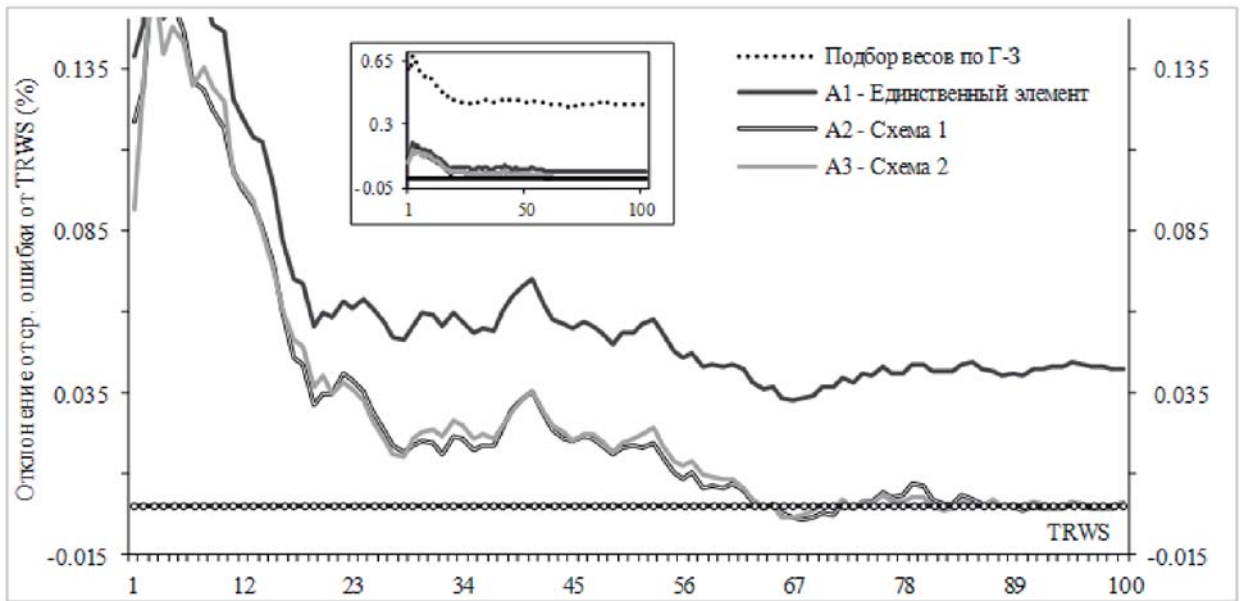


Рисунок 4.31 – График отклонения средних чисел ошибок от TRWS при обычном порядке просмотра изображений

**Второй случай:**  $(s_1, s_2, \dots, s_{100})$  представляет собой обратный порядок индексов изображений, т.е.  $(s_1, s_2, \dots, s_{100}) = (100, 99, \dots, 1)$ .

Тогда соответствующие графики таких линий средних чисел ошибок (в процентах) приведены ниже (рис. 4.32).

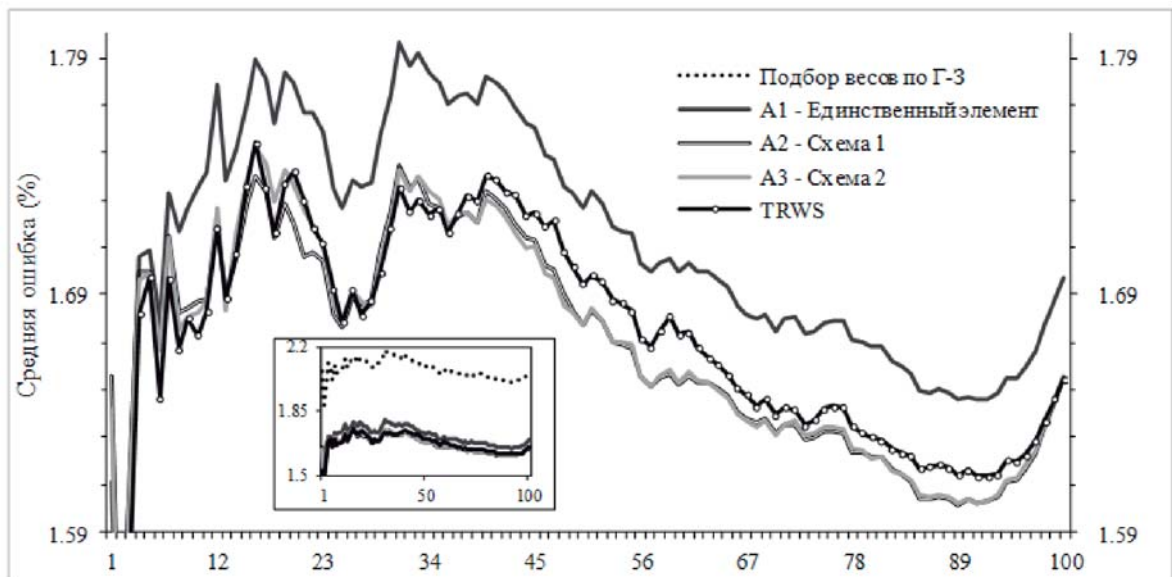


Рисунок 4.32 – График линий средних чисел ошибок при обратном порядке просмотра изображений

Если мы рассматриваем те же изображения в обратном порядке просмотра, то мы опять получим аналогичные кривые (рис. 4.32). Здесь алгоритм



TRWS уступает двум схемам подбора со многими элементами по качеству средней ошибки. Это можно увидеть из графика отклонения средней ошибки от алгоритма TRWS (рис. 4.33), где на большом интервале линии двух схем подбора со многими элементами находятся ниже линии алгоритма TRWS.

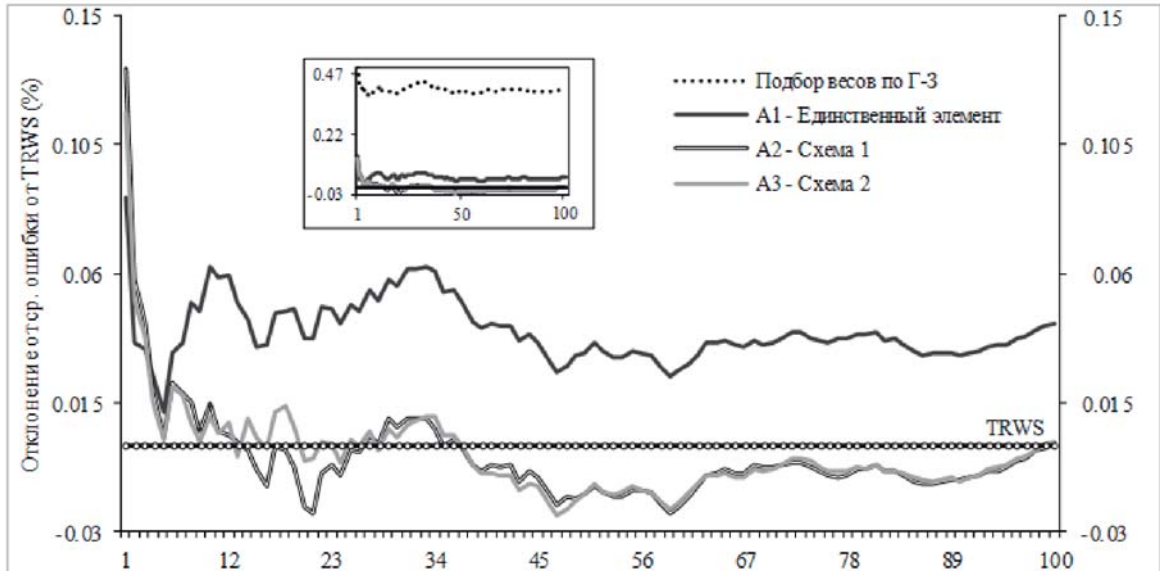


Рисунок 4.33 – График отклонения средних чисел ошибок от TRWS при обратном порядке просмотра изображений

**Третий случай:**  $(s_1, s_2, \dots, s_{100})$  представляет собой случайный порядок индексов изображений, например,  $(s_1, s_2, \dots, s_{100}) =$

(73 57 49 40 98 81 51 48 71 29 32 93 41 47 96 13 63 20 9  
75 64 4 56 14 68 65 2 38 79 39 87 21 5 46 82 85 59 66 15  
61 99 72 78 67 86 18 45 62 89 33 7 74 53 94 36 97 42 16  
55 28 70 77 12 22 25 27 90 100 58 50 44 88 60 17 76 11 19  
34 35 6 91 37 84 80 30 52 83 23 8 31 3 92 1 69 24 95 54  
26 10 43)

Тогда соответствующий график таких линий средних чисел ошибок (в процентах) приведен ниже (рис. 4.34).

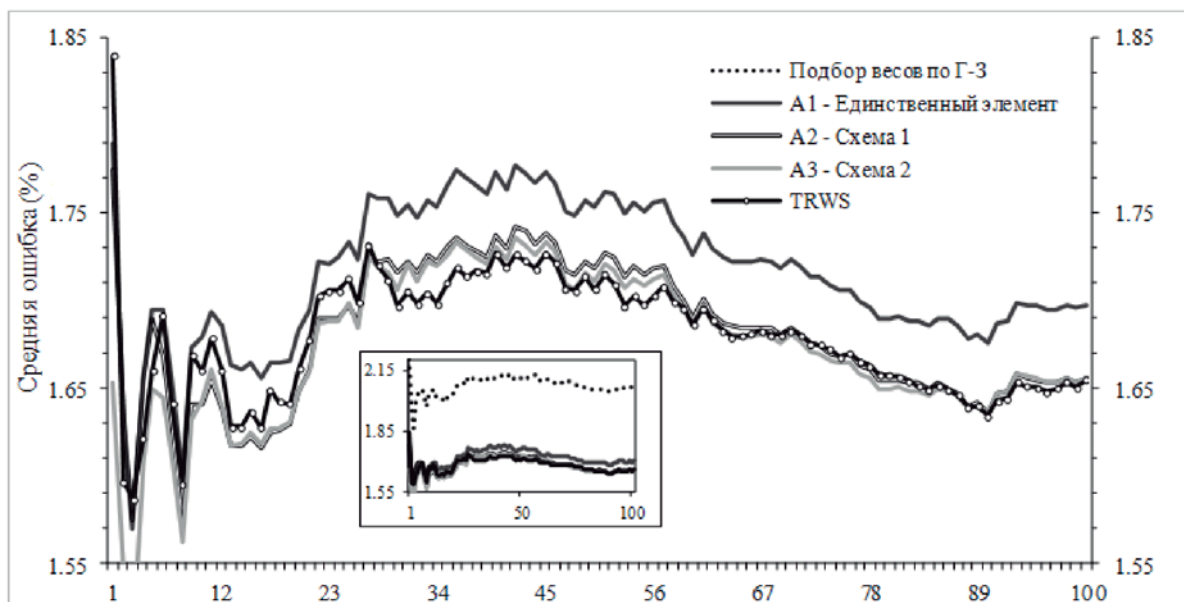


Рисунок 4.34 – График линий средних чисел ошибок при случайном порядке просмотра изображений

График отклонения средней ошибки от алгоритма TRWS показан на рис. 4.35.

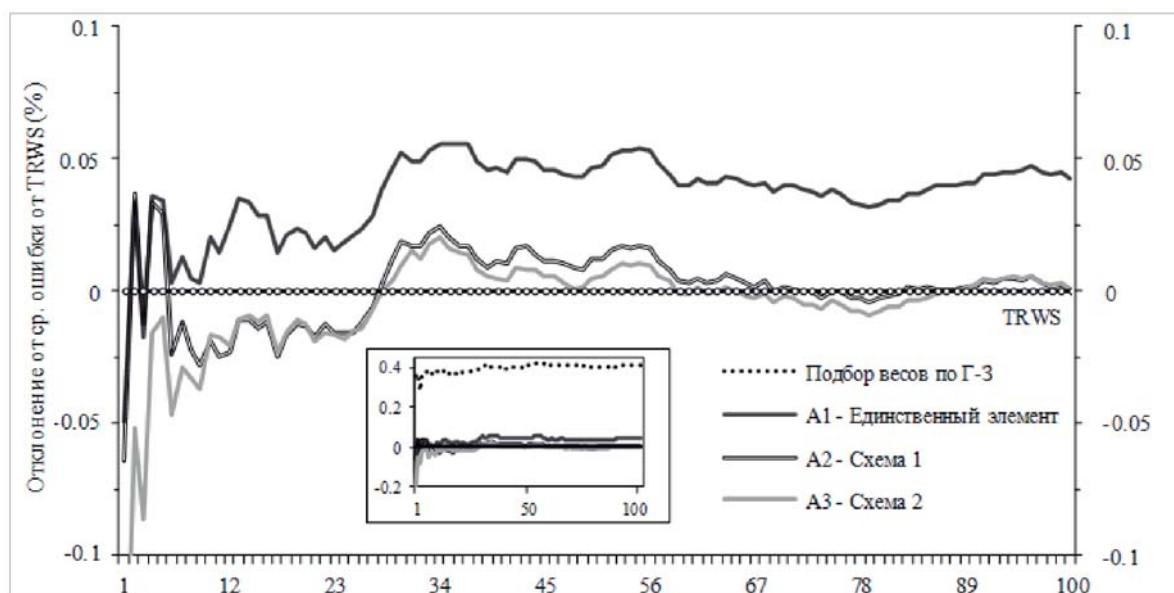


Рисунок 4.35 – График отклонения средних чисел ошибок от TRWS при случайном порядке просмотра изображений

На рис. 4.34–4.35 мы видим то же самое. Есть моменты, когда алгоритм TRWS уступает, есть моменты, когда он лучше, и есть моменты, когда уро-

вень ошибок примерно одинаковый. Это означает, что эти схемы со многими элементами  $q$  по качеству не уступают алгоритму TRWS.

Если рассмотрим отдельно отклонение ошибки от алгоритма TRWS на каждом изображении, то линии таких отклонений приведены на рис. 4.36.

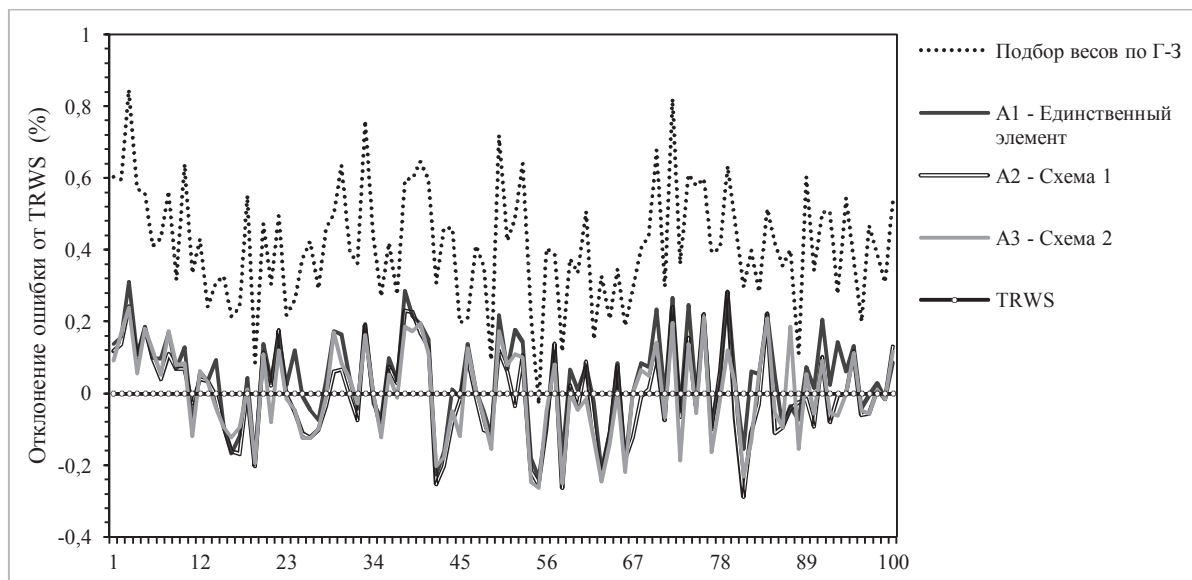


Рисунок 4.36 – Отклонение ошибки от TRWS (%)

На рис. 4.36 видно, что, хотя по среднему числу ошибок алгоритм подбора параметров с единственным диагональным элементом почти всегда уступает алгоритму TRWS (рис. 4.30–4.35), но существуют многие изображения, на которых он в действительности выигрывает у алгоритма TRWS. Алгоритм подбора весов по схеме Гаусса-Зайделя также один раз выигрывает у алгоритма TRWS. Результаты экспериментов показывают, что алгоритм TRWS проигрывает алгоритму подбора параметров с единственным элементом 35 раз, в первой схеме подбора параметров со многими диагональными элементами – 55 раз, во второй схеме со многими элементами – 50 раз.

Таким образом, разработанные алгоритмы подбора параметров комбинирования ациклических графов позволяют резко улучшить качество распознавания по сравнению с исходным алгоритмом подбора параметров по схеме Гаусса-Зайделя, в котором элемент  $q$  не подбирается и задается эвристически.

Разработанные в данной диссертации алгоритмы подбора параметров комбинирования графов соседства сравнимы по качеству распознавания с алгоритмом TRWS, который сегодня считается одним из наиболее эффективных алгоритмов распознавания.

#### 4.5 Проверка статистической гипотезы о равенстве средних уровня ошибок распознавания

Согласно [6], пусть генеральные совокупности  $X, Y$  распределены нормально, причем их дисперсии известны. По независимым выборкам объема  $n$  найдены выборочные средние  $\bar{x}, \bar{y}$ .

Требуется по выборочным средним при заданном уровне значимости  $\alpha$  проверить нулевую гипотезу, состоящую в том, что генеральные средние рассматриваемых совокупностей равны между собой, т.е.  $H_0 : M(X) = M(Y)$ .

Пусть конкурирующая гипотеза есть  $H_1 : M(X) \neq M(Y)$ . Тогда критическая область является двухсторонней, которая определяется критическим значением  $z_{кр} = \Phi^{-1}((1 - \alpha) / 2)$ , где  $\Phi^{-1}$  – обратное преобразование Лапласа.

В качестве критерия проверки нулевой гипотезы применяется случайная величина:

$$Z = \frac{\bar{X} - \bar{Y}}{\sigma(\bar{X} - \bar{Y})} = \frac{\bar{X} - \bar{Y}}{\sqrt{D(X) / n + D(Y) / n}}$$

Пусть значение критерия, вычисленного по данным наблюдений, обо-

значается как  $Z_{набл}$ , где  $Z_{набл} = \frac{\bar{x} - \bar{y}}{\sqrt{D(X) / n + D(Y) / n}}$ .

Тогда, если  $|Z_{набл}| < z_{кр}$  – нулевая гипотеза не отвергается.

А если  $|Z_{набл}| > z_{кр}$  – нулевая гипотеза отвергается.

Согласно [6] если независимая выборка имеет большой объем (не менее 30), то выборочное среднее распределено приблизительно нормально, а выборочные дисперсии являются достаточно хорошими оценками генеральных дисперсий и в этом смысле их можно считать известными приближенно.

Поэтому, если генеральные совокупности не распределены нормально и их дисперсии неизвестны, причем выборки имеют большой объем и независимы, то можно сравнивать их средние так, как описано выше, заменив точный критерий  $Z$  приближенным критерием  $Z'$ , где

$$Z' = \frac{\bar{X} - \bar{Y}}{\sqrt{D_B(X) / n + D_B(Y) / n}}.$$

В этом случае наблюдаемое значение приближенного критерия таково:

$$Z'_{набл} = \frac{\bar{x} - \bar{y}}{\sqrt{D_B(X) / n + D_B(Y) / n}}.$$

Рассмотрим число ошибок распознавания изображений, полученное каждым алгоритмом, как соответствующую случайную величину.

Подбор весов соответствует случайной величине  $weval\_E$ .

Подбор диагонального элемента и весов графов соответствует случайной величине  $single\_E$ .

Первая схема подбора параметров со многими диагональными элементами соответствует случайной величине  $first\_E$ .

Вторая схема подбора параметров со многими диагональными элементами соответствует случайной величине  $second\_E$ .

Алгоритм TRWS соответствует случайной величине  $trws\_E$ .

Числа ошибок распознавания, полученные каждым алгоритмом на всех изображениях, составляют генеральную совокупность числа ошибок, которая соответствует данному алгоритму.

Были сгенерированы 100 текстурных изображений. Поэтому по каждому алгоритму распознавания получаются 100 чисел ошибок распознавания, которые составляют соответствующую данному алгоритму выборку большого объема из генеральной совокупности чисел ошибок распознавания.

По результатам распознавания вычисляются следующие выборочные средние и выборочные дисперсии.

$$M_B(weval\_E) = 2.0667063; D_B(weval\_E) = 0.1633142$$

$$M_B(\text{single}_E) = 1.6972105; D_B(\text{single}_E) = 0.1228320$$

$$M_B(\text{first}_E) = 1.6550333; D_B(\text{first}_E) = 0.1177515$$

$$M_B(\text{second}_E) = 1.6556768; D_B(\text{second}_E) = 0.1214256$$

$$M_B(\text{trws}_E) = 1.6546125; D_B(\text{trws}_E) = 0.1096015$$

Для проверки гипотезы о равенстве средних уровня ошибок распознавания каждой парой алгоритмов, сначала вычислим наблюдаемое значение критерия сравнения.

$$\text{Подбор весов и TRWS: } Z'_{\text{набл}} = 7.8882694.$$

$$\text{Подбор диагонального элемента и весов графов и TRWS: } Z'_{\text{набл}} = 0.8835667.$$

$$\text{Первая схема со многими элементами и TRWS: } Z'_{\text{набл}} = 0.0088248.$$

$$\text{Вторая схема со многими элементами и TRWS: } Z'_{\text{набл}} = 0.0221434.$$

$$\text{Если уровень значимости } \alpha = 0.05, \text{ то } z_{\text{кр}} = 1.96.$$

В итоге, при уровне значимости  $\alpha = 0.05$ , гипотеза о равенстве средних уровня ошибок распознавания алгоритма подбора весов и TRWS отвергнута, а гипотеза для всех алгоритмов с подбором диагонального элемента и TRWS не отвергнута.

Таким образом, решая локальную задачу минимизации ошибок распознавания на основе численного оценивания апостериорных вероятностей скрытых классов в элементах массива данных, мы построили алгоритмы того же класса по качеству распознавания, что и TRWS, который решает глобальную задачу минимизации гиббсовской энергии связей.

#### **4.6 Распознавание неподходящих изображений**

В приведенной частной модели марковского случайного поля, мы сделали ряд предположений. В этой модели марковское случайное поле выражается как эргодическая марковская цепь. Кроме этого, финальное распределение скрытых классов считали равномерным. Очевидно, что существует множество изображений, которые не подходят к нашей модели. Например, изображения на рис. 4.37. На изображении на рис. 4.37а марковская цепь нико-

гда не переходит с первого состояния в третье. А на изображении на рис. 4.37б область одного класса значительно меньше области других. Это означает, что предположение о равномерности финального распределения совсем не годится для него.

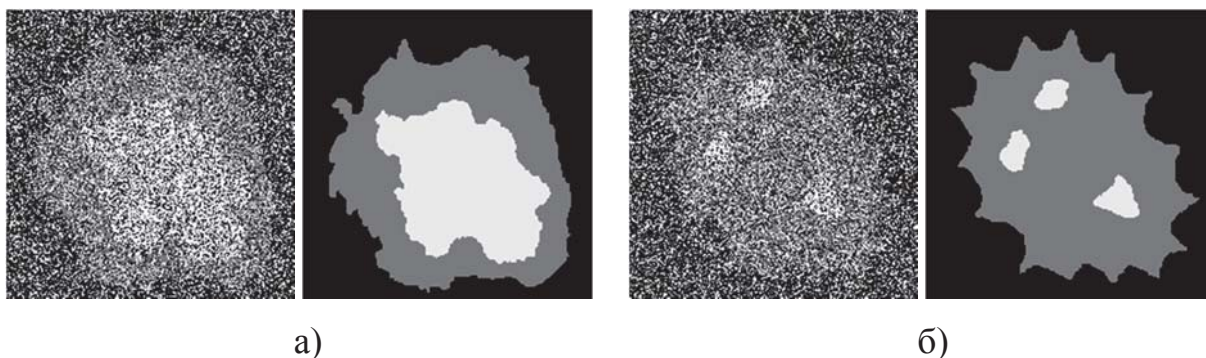


Рисунок 4.37 – Неподходящие изображения: а) неэргодический случай;  
б) неравномерное финальное распределение

Попробуем применить разработанные алгоритмы комбинирования ациклических графов на таких изображениях.

Результаты распознавания приведены на рис. 4.38.

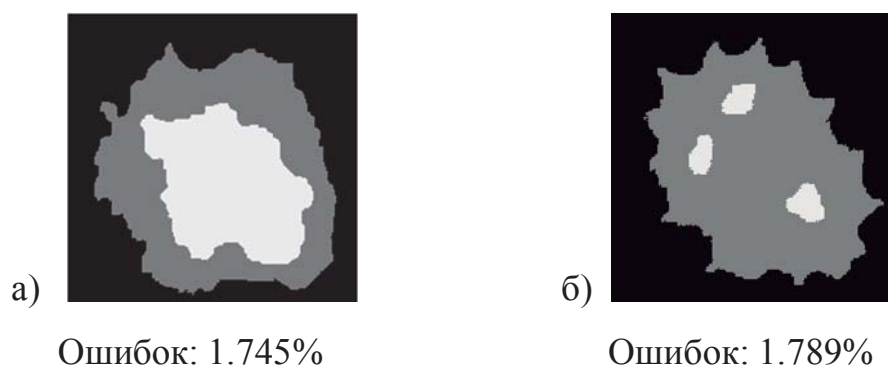


Рисунок 4.38 – Результаты распознавания на неподходящих изображениях:

а) неэргодический случай; б) неравномерное финальное распределение

Из рис. 4.38 видно, что рассмотренная частная модель марковского случайного поля эффективно работает даже на неподходящих к ней изображениях.

## 5 ОЦЕНКА ОШИБКИ РАСПОЗНАВАНИЯ МЕТОДОМ СКОЛЬЗЯЩЕГО КОНТРОЛЯ

### 5.1 Упрощенная схема скользящего контроля

Как ранее сказано, в данной работе мы предложим «гибкий» подход, который заключается в подборе марковских параметров как натуральных, но качество решающего правила проверяется на основе схемы скользящего контроля, примененной при настройке структурных параметров.

Как правило, исходная совокупность изображений разделяется на обучающее и тестовое множества, где размер тестового множества обычно составляет 25% общего объема исходной совокупности [19, 21]. Такое разбиение исходной совокупности обычно повторяется 100 раз. При повторении разбиения некоторые изображения могут ни разу не участвовать в тесте и некоторые могут ни разу не участвовать в обучении. Для справедливости в упрощенной схеме скользящего контроля каждый раз мы будем оставлять только одно изображение в качестве тестового, а остальные изображения служат для выполнения схемы скользящего контроля. Таким образом, гарантируется, что каждое изображение используется как тестовое ровно один раз и ни одного изображения не участвует в обучении.

Для снижения вычислительной трудоемкости мы будем применять неклассическую схему скользящего контроля, т.е. схему *LPOCV* с  $p = N - 1$ . Это означает, что при обучении используется лишь одно изображение. Известно, что относительно малый объем обучающей совокупности ведет к переобучению. Тогда возникает вопрос, который состоит в том, достаточно ли использовать одно изображение для обучения. С одной стороны одно изображение может рассматривать как один целый объект. Такая точка зрения действительно вызывает ощущение малости объема обучающей совокупности. Но с другой стороны, изображение размера  $201 \times 201$  состоит из примерно 40 тысяч пикселей. Если каждая пиксель рассматривается как один объект, то использование одного изображения для обучения на самом деле озна-



чает выполнение обучения около на 40 тысяч объектов. Это число, пожалуй, достаточно большое.

Кроме того, все изображения генерируются случайным образом на основе одних и тех же распределений. Поэтому характеры их текстурных классов почти одинаковы. Они отличаются лишь формой границ, которая вносит основной вклад в эффект переобучения при обучении на одном изображении. Однако минимизация ошибок внутри классов, безусловно, важнее, чем минимизация ошибок на границах. Поэтому надеемся, что переобучение, исходящее от формы границы, несильно ухудшает результат распознавания новых изображений.

Другое обоснование, позволяющее свести классическую схему к неклассической схеме скользящего контроля, заключается в том, что результаты подбора марковского параметра на одном изображении и на большинстве изображений несильно отличаются между собой. Для выяснения этого мы сделаем следующие эксперименты. Берется последовательно каждое изображение. На нем строится линия ошибок распознавания для изменяющегося значения диагонального элемента. Строится аналогичная линия средних ошибок на остальных без него самого. Эксперименты показывают, что значения диагонального элемента, доставляющие минимум ошибка в обоих случаях почти совпали.

Линии ошибок (в логарифмическом масштабе), например, на первом изображении (сплошная линия) и на 99 остальных (пунктирная линия) для случая ступенчатого дерева приведены на рис. 5.1.

Линии ошибок (в логарифмическом масштабе), например, на первом изображении (сплошная линия) и на 99 остальных (пунктирная линия) для случая комбинирования ациклических графов соседства с равными весами приведены на рис. 5.2.

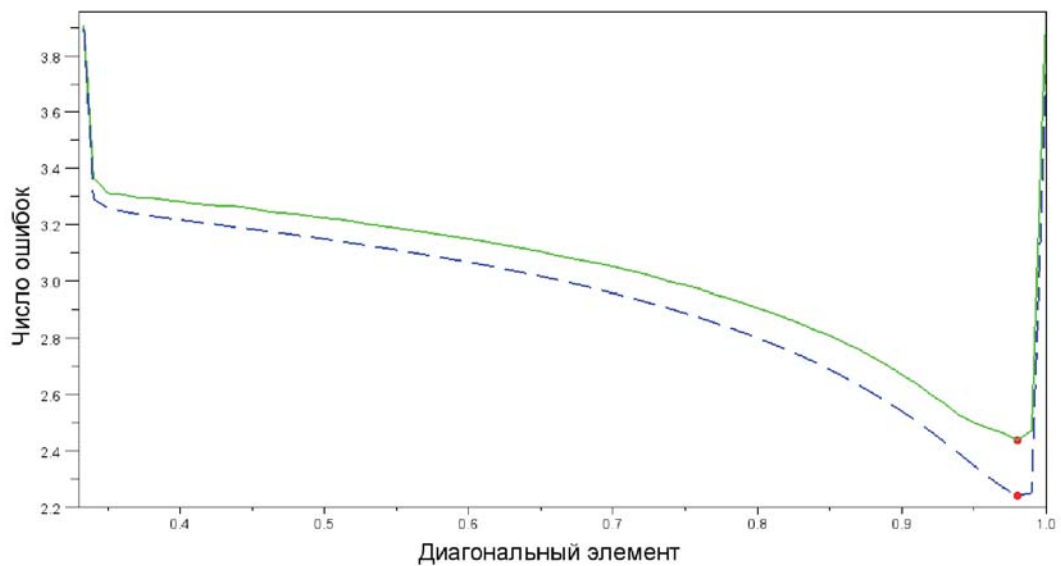


Рисунок 5.1 – Линии ошибок распознавания (в логарифмическом масштабе) на первом и остальных изображениях для случая ступенчатого дерева.

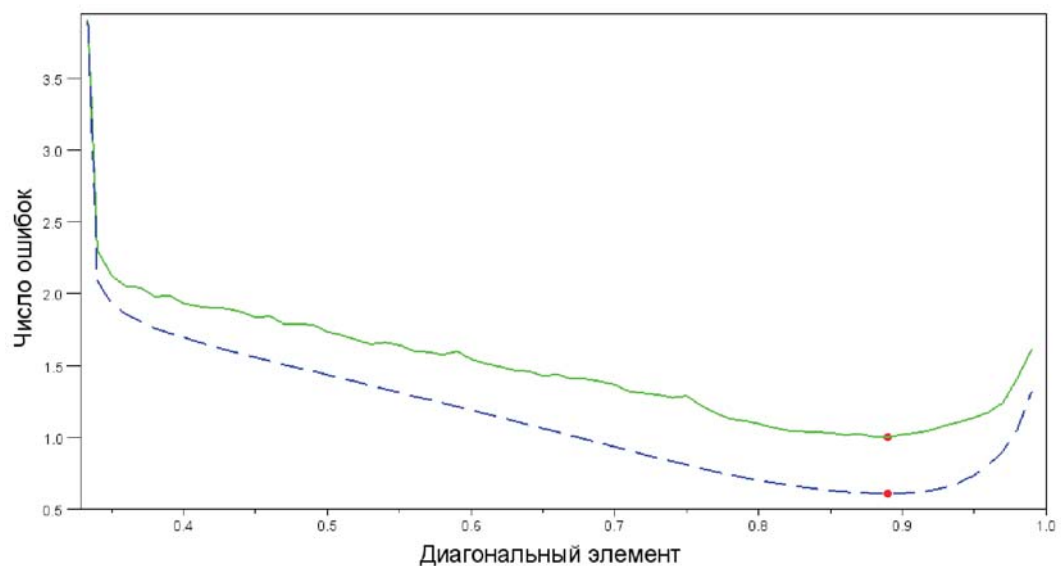


Рисунок 5.2 – Линии ошибок распознавания (в логарифмическом масштабе) на первом и остальных изображениях для случая комбинирования ациклических графов.

Вышесказанные соображения позволяют построить упрощенную схему скользящего контроля для оценки ошибки на генеральной совокупности, суть которой заключается в следующем.

1. Задать совокупность изображений.

2. Исключить очередное изображение и использовать его в качестве тестового.

3. Остальные изображения составляют обучающее множество. На нем выполнить неклассическую схему скользящего контроля:

- выбрать очередное изображение из обучающего множества. Построить для него решающее правило распознавания;
- остальные изображения в обучающем множестве использовать для валидации построенного решающего правила и оценить среднее число ошибок валидации;
- найти набор параметров, доставляющий минимальное среднее число ошибок валидации.

4. С помощью полученного набора параметров распознать тестовое изображение.

5. Усреднить тестовую ошибку. Эта ошибка считается ошибкой распознавания на генеральной совокупности.

На практике, если дана совокупность изображений и требуется лишь дать оценку оптимальных значений параметров, то можно построить более простую процедуру без тестовых изображений. Вся данная совокупность изображений используется для обучения. Последовательно выполняется обучение на очередном изображении. На остальных изображениях выполняется распознавание и вычисляется средняя ошибка валидации. Та оценка параметров на одном изображении, которой соответствует минимальная средняя ошибка валидации на остальных, считается наиболее хорошей и целесообразной для распознавания новых поступающих изображений.

## **5.2 Алгоритмы подбора диагонального элемента по ациклическому графу**

Исключим очередное изображение для тестирования. На остальных изображениях выполним неклассическую схему скользящего контроля. Определим набор параметров, который дает наименьшую валидную ошибку.

Потом распознаем тестовое изображение. После того, усредним тестовую ошибку.

Выполним эту процедуру для пяти ациклических графов (см. рис. 1.4). Результаты проверки показаны на рис. 5.3.

На рис. 5.3 видно, что по результатам скользящего контроля алгоритм подбора диагонального элемента на основе независимого обучения по-прежнему является наихудшим.

Раньше по результатам распознавания на индивидуальных изображениях алгоритм подбора диагонального элемента по схеме Гаусса-Зайделя выиграл у всех остальных по качеству средней ошибки. Однако по результатам скользящего контроля алгоритм по схеме Гаусса-Зайделя проигрывает алгоритму подбора, основанного на заданном значении диагонального элемента  $q = 0.95$ , хотя и немного.

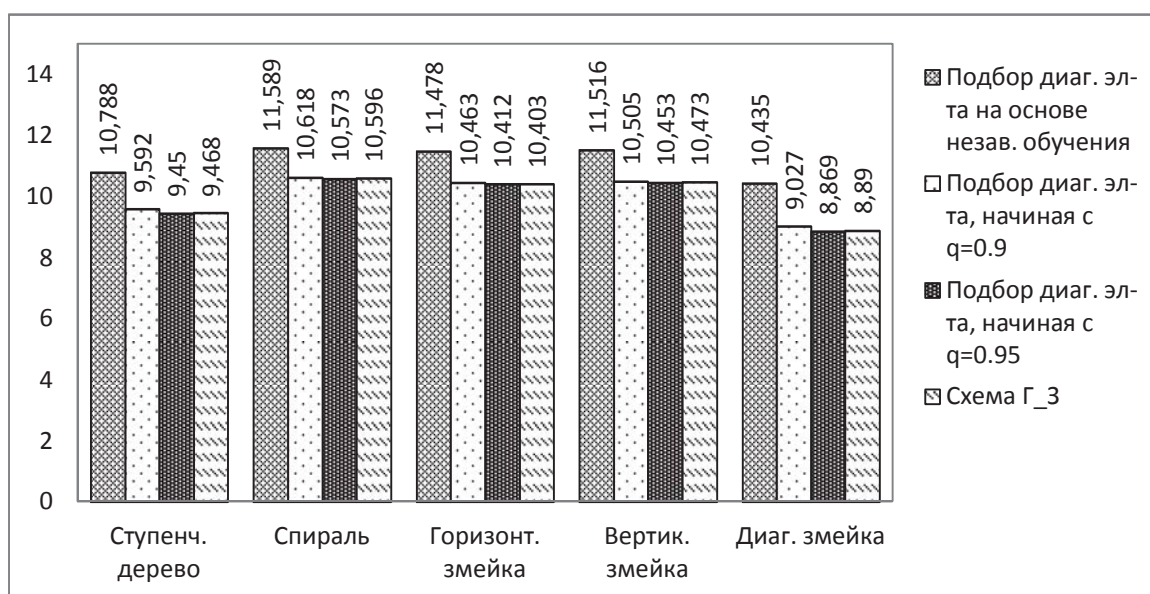


Рисунок 5.3 – Средняя ошибка распознавания (%) на 100 изображениях при неклассической схеме скользящего контроля

### 5.3 Алгоритмы подбора параметров комбинирования ациклических графов соседства

Исключим очередное изображение для тестирования. На остальных изображениях выполним неклассическую схему скользящего контроля. Определим набор параметров, который дает наименьшую валидную ошибку.

Потом распознаем тестовое изображение. После того, усредним тестовую ошибку.

Результаты оценки средней ошибки показаны на рис. 5.4.

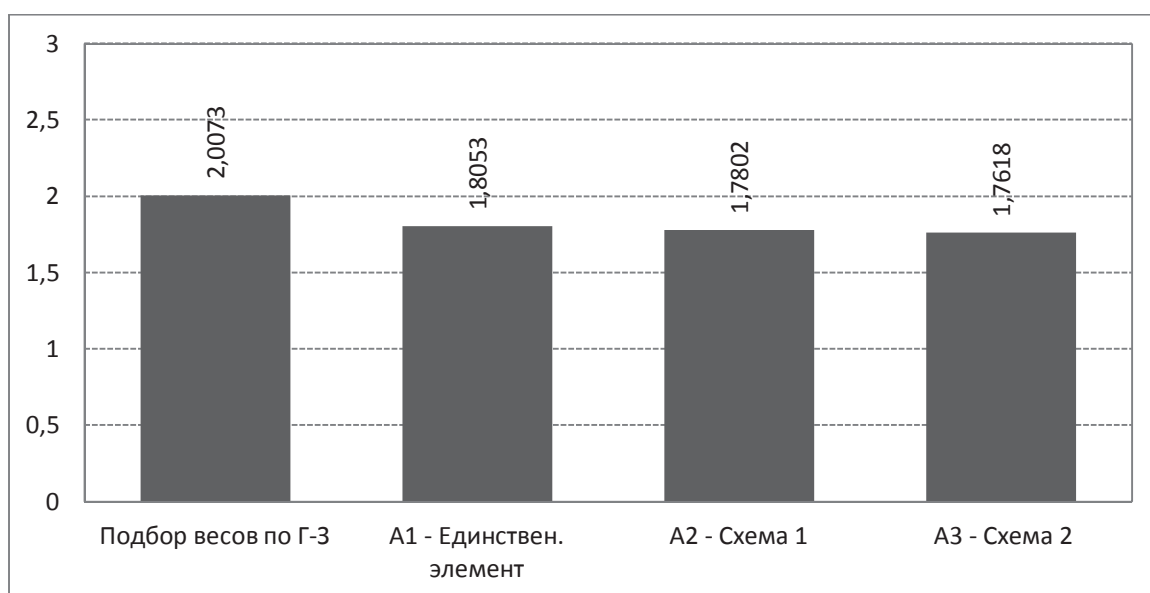


Рисунок 5.4 – Средняя ошибка распознавания (%) на 100 изображениях при неклассической схеме скользящего контроля

Результаты скользящего контроля показывают, что алгоритм подбора весов по схеме Гаусса-Зайделя является наихудшим. Алгоритм подбора единственного диагонального элемента и весов графов дает решающее правило, качество которого хуже, чем две схемы подбора параметров со многими элементами.

## 6 РАСПОЗНАВАНИЕ РЕАЛЬНЫХ ИЗОБРАЖЕНИЙ

### 6.1 Интерактивная схема выбора данных учителя

Для реальных изображений, как правило, необходимо использовать более сложный алгоритм независимого обучения для их предобработки, т.к. на реальных изображениях текстуры часто более неоднородны, и их цветковые гистограммы обычно не распределены по одному нормальному распределению. Поэтому оценка параметров по формуле (4.2) на основе принципе максимального правдоподобия с одним нормальным распределением (4.1) обычно не дает достаточно хорошего результата, который мог бы отразить сущность цветкового распределения текстур на исходном изображении.

Для предобработки реальных изображений часто применяется модель смеси гауссовских распределений с EM-алгоритмом [29, 32, 36, 40, 46]. Однако результат независимого обучения сильно зависит от выбора информативных признаков и выборочных данных учителя. Неподходящий выбор информативных признаков и плохой выбор данных учителя приводят к плохому результату независимого обучения, который, в свою очередь, приводит к плохому результату распознавания. Это объясняется тем, что плохой результат независимого обучения часто содержит грубые ошибки, которые алгоритмы распознавания не всегда могут исправить.

При обработке изображений существует ряд информативных признаков [30]. Задача выбора подходящих к типу обрабатываемых изображений признаков очень важна [26]. В данной работе задача выбора информативных признаков не рассматривается. Для упрощения в качестве информативных признаков будем использовать цветковые компоненты RGB [41] точек изображений. Очевидно, что такие наблюдаемые признаки не подходят к массивам реальных изображений. Например, при обработке черно-белых изображений, представленных в серой шкале [41], выбор цветковых компонент RGB точек изображений в качестве информативных признаков естественно не пригодится, т.к. они имеют одинаковые значения. Для таких изображений необходимо использовать другие информативные признаки, например,

фильтр Габора [28, 33], яркость, дисперсию яркости в окне, преобразование Фурье [5, 30, 41] и т.п.

Чтобы получить достаточно хороший результат независимого обучения мы предложим интерактивную схему выбора данных учителя, которая состоит из следующих шагов.

1) Пусть задано множество  $m$  классов  $\Omega = \{1, 2, \dots, m\}, m \geq 2$ . Выберем случайную область объектов  $U_i = U_{i0}$  для каждого класса  $\omega_i$ , где  $U_{i0}$  обозначает начальную выборочную область для класса  $\omega_i, i = 1, \dots, m$ . Здесь  $U_i, i = 1, \dots, m$  понимаются как выборочные совокупности объектов.

2) Выполним алгоритм независимого обучения на основе выборочных совокупностей объектов  $U_i, i = 1, \dots, m$ .

3) Если в результате независимого обучения существуют нежелательные области объектов, т.е. те области, в которых большинство объектов неправильно распознавались, то следует обновить выборочные области учителя  $U_i, i = 1, \dots, m$  по следующему правилу.

По очереди рассмотреть каждую нежелательную область. Пусть нежелательная область  $U_{Err}$  фактически принадлежит классу  $\omega_i$ . Тогда необходимо добавить в выборочную совокупность  $U_i$  объекты из подобласти  $U_{Err}^* \subset U_{Err}$ , где подобласть  $U_{Err}^*$  выбирается так, чтобы она могла бы отразить характеристику цветового распределения всей нежелательной области  $U_{Err}$ . Таким образом:  $U_i = U_{i(prev)} \cup U_{Err}^*$ , где  $U_{i(prev)}$  – выборочная совокупность для класса  $\omega_i$  на предыдущем шаге.

4) Перейдем к шагу 2.

5) Повторим шаги 2–4 до тех пор, пока нежелательные области не исчезнут.

Продemonстрируем интерактивную схему выбора данных учителя, например, на изображении побережья Мексиканского залива (рис. 6.1 [54]).



Рисунок 6.1 – Побережье Мексиканского залива

Наша цель – разделить изображение на два класса: «земля» и «море». Пусть сначала выберем выборочные совокупности как показано на рис. 6.2а.

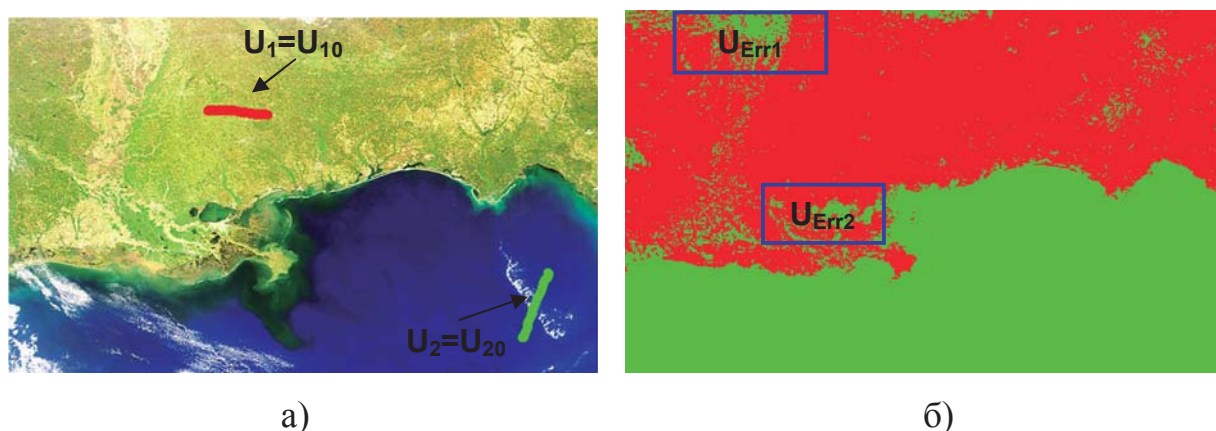


Рисунок 6.2 – Первое выполнение независимого обучения: а) Начальный выбор областей данных учителя; б) Результат обучения

На результирующем изображении (рис. 6.2б) появилась нежелательные области  $U_{Err1}$  и  $U_{Err2}$ , которые показаны прямоугольниками. Эти области должны принадлежать классу «земля», но по результату обучения большинство их объектов принадлежит классу «море».

По интерактивной схеме необходимо выделить области текстур класса «земля»  $U_{Err1}^*$ ,  $U_{Err2}^*$  из нежелательных областей  $U_{Err1}$ ,  $U_{Err2}$  так, чтобы каждая область  $U_{Err_i}^*$  могла бы отразить характеристику цветового распределения области  $U_{Err_i}$ ,  $i=1,2$ . Потом необходимо добавить области  $U_{Err_i}^*$ ,  $i=1,2$  в выборочную совокупность  $U_1$  класса «земля». Например, выделим области  $U_{Err1}^*$ ,



$U_{Err2}^*$  как показано на рис. 6.3а. Снова выполним независимое обучение. Результат приведен на рис. 6.3б.

Как видно из рис. 6.3б, в новом результате обучения появились новые нежелательные области, которые должны принадлежать классу «море». Согласно предложенной интерактивной схеме выбора данных учителя, опять необходимо обновить выборочные совокупности добавлением новых объектов в выборочную совокупность  $U_2$  класса «море», например, как показано на рис. 6.4а.

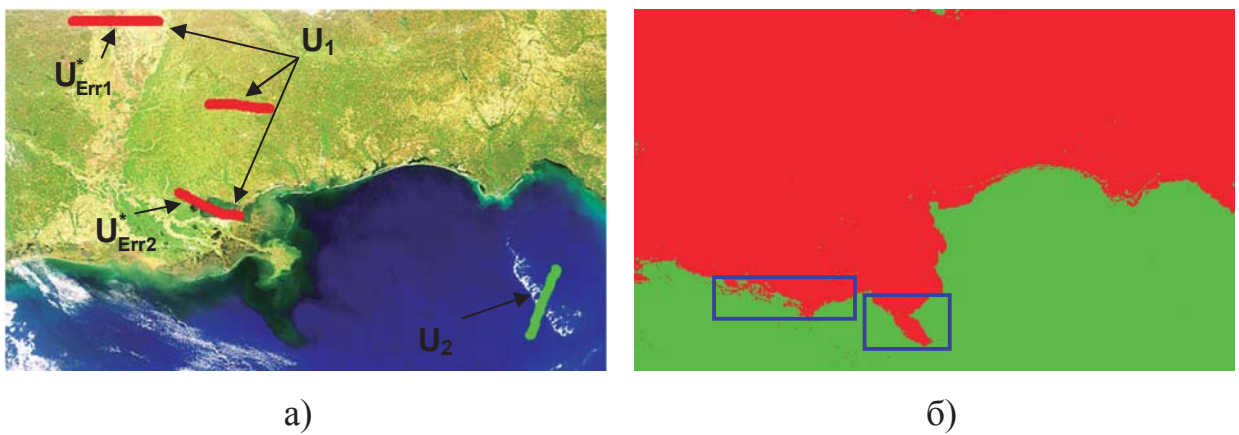


Рисунок 6.3 – Второе выполнение независимого обучения: а) Начальный выбор областей данных учителя; б) Результат обучения

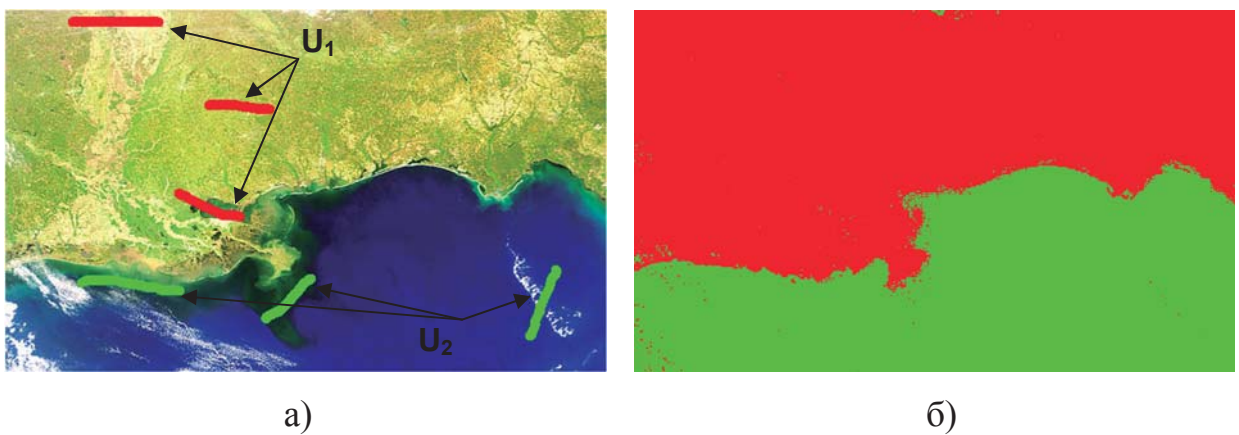


Рисунок 6.4 – Третье выполнение независимого обучения: а) Начальный выбор областей данных учителя; б) Результат обучения

Опять выполним независимое обучение. И получим новый результат как показано на рис. 6.4б. На рис. 6.4б мы не видим ни одной нежелательной

области. Поэтому можно остановить процедуру интерактивного независимого обучения.

Тем не менее, условие остановки такой интерактивной схемы выбора данных учителя не всегда выполняется. Эта схема не сходится, когда существует хотя бы одна нежелательная область  $U_{Err}$  в одном классе текстуры  $i \in \Omega$ , цветовая характеристика которой слишком похожа на цветовую характеристику другого класса  $j \in \Omega$  (точнее одной части другого класса). В этом случае, несмотря на то, что мы добавляем в выборочную совокупность  $U_i$  объекты из корректирующей области  $U_{Err}^* \subset U_{Err}$ , алгоритм всегда решает, что область  $U_{Err}$  принадлежит классу  $j$ . Это объясняется тем, что добавляемые объекты из корректирующей области  $U_{Err}^* \subset U_{Err}$  мало влияют на результат EM-алгоритма. Поэтому после обучения смесь гауссовских распределений у класса  $i$  остается менее подходящей к объектам нежелательной области  $U_{Err}$ , чем смесь гауссовских распределений у класса  $j$ .

Подробнее рассмотрим изображение «Утки» (рис. 6.5 [59]). Наша цель – выделить «утки» из фона.



Рисунок 6.5 – Изображение «Утки»

Первое выделение выборочных совокупностей (рис. 6.6а) оказывается не хорошим, так как в результате обучения возникают многие нежелательные области (рис. 6.6б).

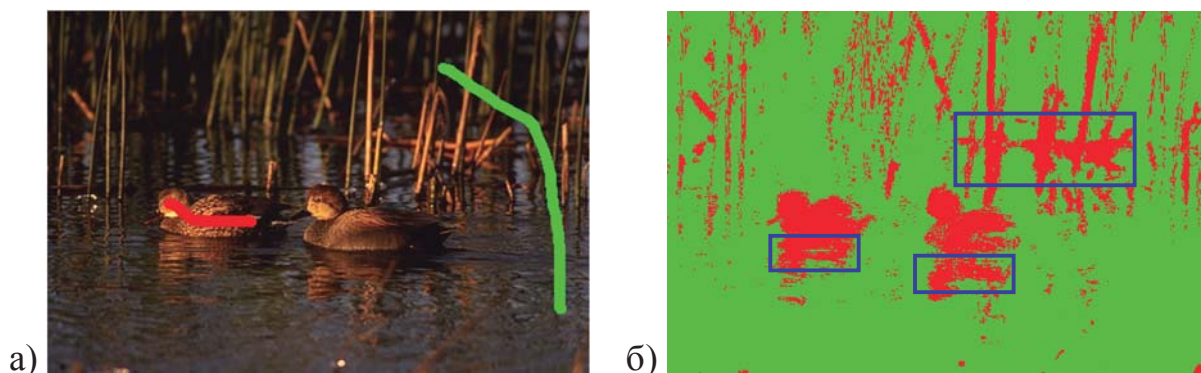


Рисунок 6.6 – Первое обучение: а) Начальный выбор областей данных учителя; б) Результат обучения

Попробуем удалить нежелательные области добавлением их объектов в выборочные совокупности (рис. 6.7а). Одна из трех нежелательная область остается. И одна новая нежелательная область возникает вследствие попытки обновления данных учителя (рис. 6.7б).

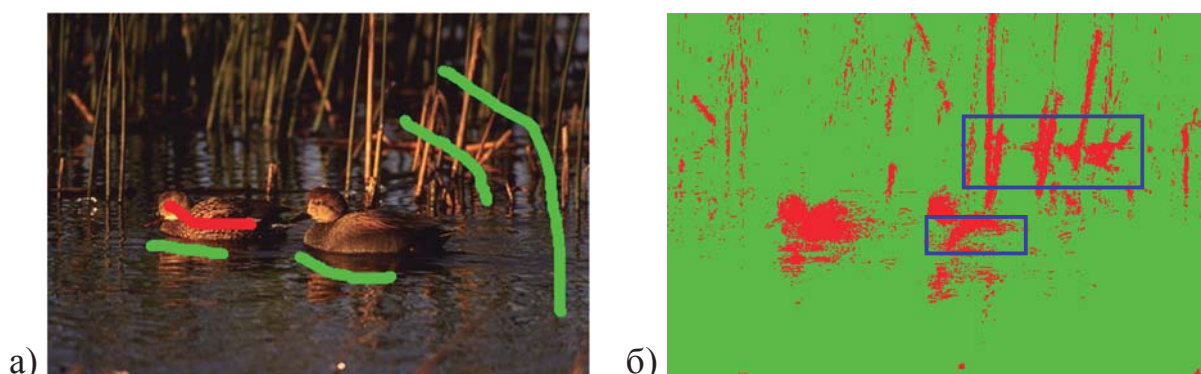


Рисунок 6.7 – Второе обучение: а) Второй выбор областей данных учителя; б) Результат обучения

Сильная корректировка старой нежелательной области (рис. 6.8а) не улучшает результат обучения. Наоборот, нежелательные области остаются и, кроме этого, возникают новые ошибочные области. Это значит, что в данном случае условие остановки схемы выбора данных учителя никогда не выполнится.

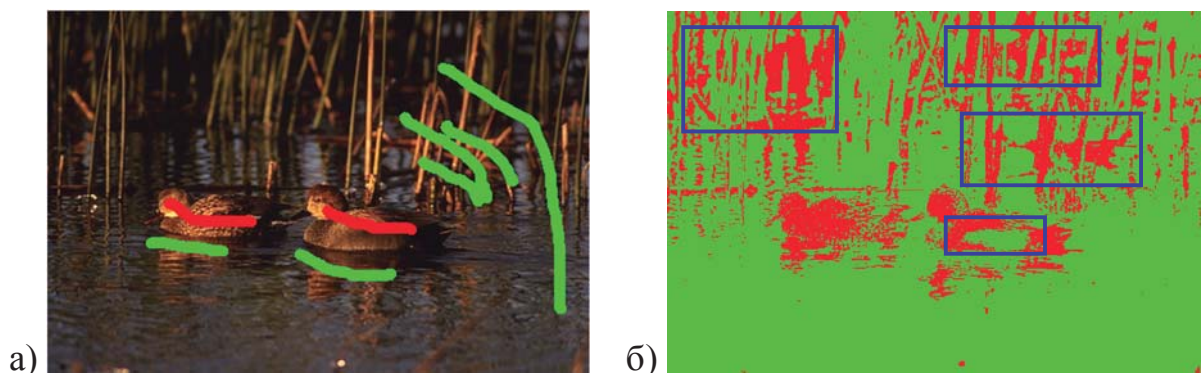


Рисунок 6.8 – Второе обучение: а) Второй выбор областей данных учителя; б) Результат обучения

## 6.2 Схема независимого обучения с использованием информации о расположении объектов классов в поле носителя

Во многих случаях интерактивная схема выбора данных учителя оказывается эффективной для независимого обучения. Однако, если текстурные классы изображения содержат области с похожими цветовыми характеристиками, то данная схема оказывается не очень эффективной. В качестве примера такого случая рассмотрим другое изображение (рис. 6.9 [57]).



Рисунок 6.9 – Изображение «Цветок»: а) Исходное изображение; б) Выбор данных учителя

Видно, что выбор темной области в цветке не позволяет удалить нежелательную область внутри цветка (рис. 6.10).



Рисунок 6.10 – Результат независимого обучения изображения «цветок»

Для решения этой проблемы предложим схему независимого обучения с использованием информации о расположении объектов классов в поле носителя, которая заключается в восстановлении априорных распределений скрытых классов за счет указанных областей учителя.

Новая схема независимого обучения отличается от алгоритмов независимого обучения тем, что в ней учитывается расположение указанных учителем областей.

Пусть задано множество  $n$  классов  $\Omega = \{1, 2, \dots, m\}$ ,  $m \geq 2$ . Новая схема независимого обучения основана на известном EM-алгоритме [29, 32, 36, 40, 46]. В качестве данных учителя для  $i$ -го класса выделяется область  $U_i$ , где  $i = 1 \dots m$ . Тогда, новая схема независимого обучения может быть описана следующим образом:

1) По данным учителя построим решающее правило на основе EM-алгоритма.

2) По полученному решающему правилу восстановим априорное распределение в тех точках, которые никакой области  $U_i$  не принадлежат.

3) Для тех точек  $t \in T$ , которые принадлежат области  $U_i$ ,  $i = 1 \dots m$ , зададим апостериорное распределение:  $p(x_t | y_t)$ , где

$$p(x_t = i | y_t) = 1, p(x_t = j | y_t) = 0, j \neq i.$$

Результат новой схемы независимого обучения с таким же выбором данных учителя (см. рис. 6.9б) оказывается лучше. Нежелательная область внутри цветка удалена (рис. 6.11).



Рисунок 6.11 – Результат новой схемы независимого обучения изображения «цветок»

### 6.3 Сравнение алгоритмов при распознавании реальных изображений

Применим разработанные алгоритмы для распознавания изображения «цветок» (рис. 6.9а). В качестве результата независимого обучения используем результат, показанный на рис. 6.11. Для оценки числа ошибок предлагается, что учитель полностью выделил цветок на изображении (рис. 6.12а).

Сначала возьмем алгоритм комбинирования ациклических графов без подбора параметров ( $q = 0,95$ ). Результат распознавания приведен на рис. 6.12б. Число ошибок составляет 3912 ошибок, т.е. 1,45%.

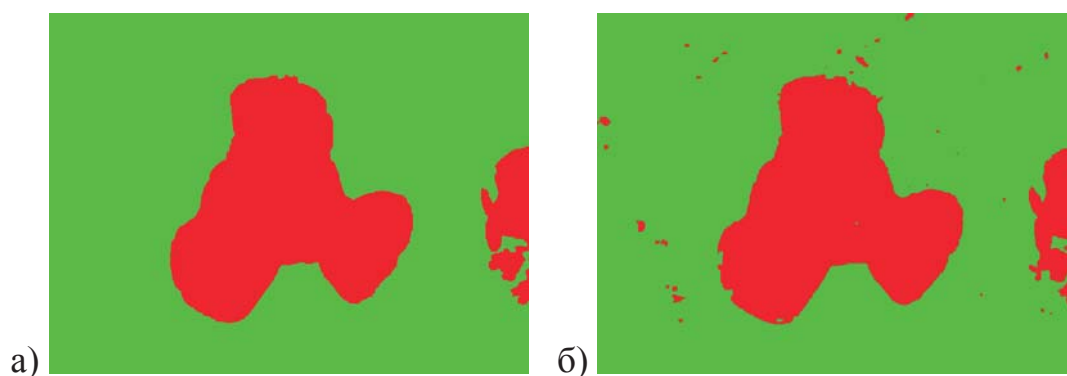


Рисунок 6.12 – Сегментация изображения: а) выделение учителем цветка ; б) результат распознавания алгоритмом комбинирования ациклических графов без подбора параметров

Предполагается, что если набор значений параметров алгоритма комбинирования ациклических графов оказывается эффективным при обработке модельных изображений, то он также будет хорошим для распознавания реальных изображений. Исходя из такого эвристического соображения, оценим параметры комбинирования ациклических графов на совокупности 100 модельных изображений по неклассической схеме скользящего контроля. Последовательно выполняется обучение на очередном изображении и получается оценка параметров комбинирования ациклических графов. На остальных изображениях выполняется распознавание и вычисляется средняя ошибка валидации. Среди полученных оценок определяется набор параметров, который дает наименьшую среднюю ошибку валидации (см. приложение А). Это есть подобранные параметры на 52-ом изображении при первой схеме со многими элементами (см. приложение Б).

Веса графов: (0.46511, 0.13, 0.14111, 0.14111, 0.12267).

Диагональные элементы: (0.37, 0.61, 0.9, 0.92, 0.82).

Результат распознавания изображения «цветок» алгоритмом комбинирования ациклических графов с подбором параметров приведен на рис. 6.13а. Число ошибок составляет 3719 ошибок, т.е. 1,377%.

Результат распознавания изображения алгоритмом TRWS приведен на рис. 6.13б. Число ошибок составляет 3891 ошибок, т.е. 1,44%.

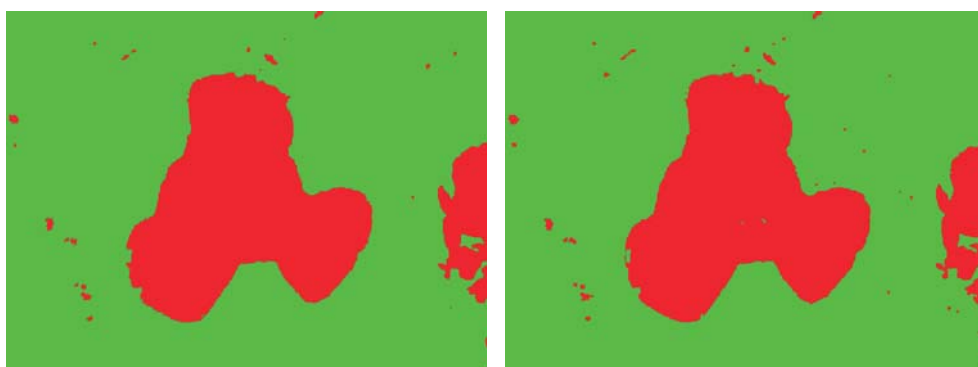


Рисунок 6.13 – Результат распознавания алгоритмом комбинирования ациклических графов: а) с подбором параметров; б) TRWS

Видно, что алгоритм комбинирования ациклических графов с подбором параметров улучшает результат по сравнению с алгоритмом без подбора, хотя не очень сильно.

Разработанный алгоритм распознавания с подбором параметров работает не хуже, чем алгоритм TRWS. Результаты обработки практически одинаковы.

#### 6.4 Обновление апостериорных маргинальных распределений

Эксперименты показывают, что комбинирование ациклических графов позволяет удалить мелкие нежелательные области. Однако процедура удаления мелких нежелательных областей эффективна только, если данные еще «хаотические», т.е. каждая точка может принадлежать тому и иному классу с достаточно большими вероятностями.

Тем не менее, при увеличении числа итераций апостериорные маргинальные вероятности быстро стремятся к нулям и единицам. Тогда повторение дополнительных итераций не поможет улучшить результат распознавания.

Рассмотрим конкретный пример на следующем изображении (рис. 6.14а [58]).



Рисунок 6.14 – Изображение «Дорога»: а) Исходное изображение; б) Выбор данных учителя


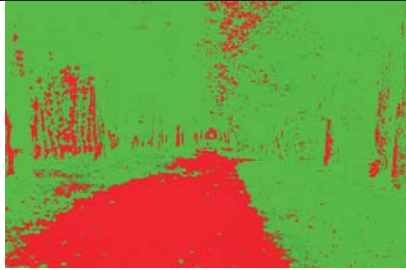
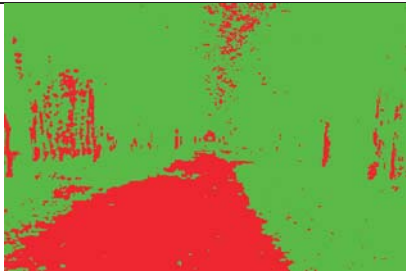
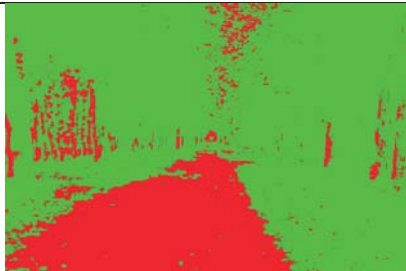
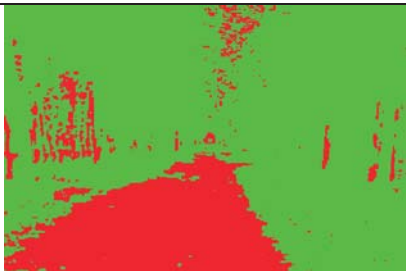
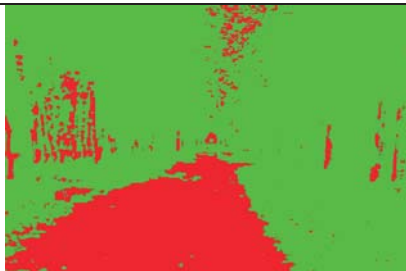
Здесь мы будем выделять дорогу (серую область) из фона. Пусть данные учителя выбираются как на рис. 6.14б. Результаты распознавания будут отображены в трех режимах:

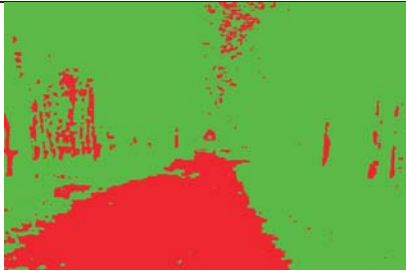
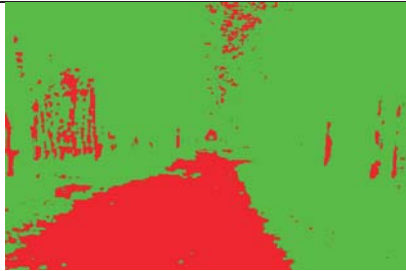
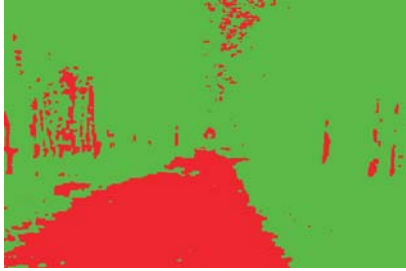
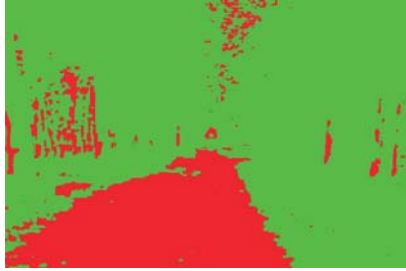
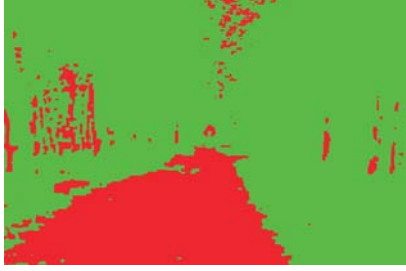









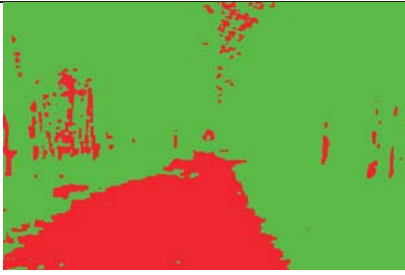
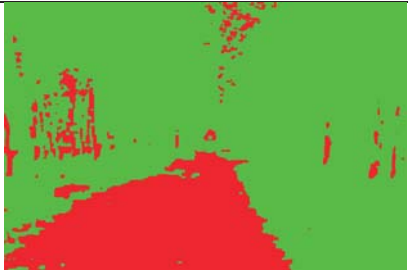

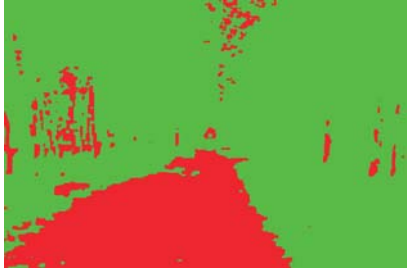
- Обычный режим: Каждая точка, принадлежащая классу «Дорога» отображена красным цветом; а точка фона отображена зеленым.
- Смешанный режим: Пусть точка  $t$  имеет апостериорные маргинальные вероятности принадлежности к классам «дорога» и «фон», равные  $p_1$  и  $p_2 = 1 - p_1$ . Тогда точка  $t$  будет отображена цветом, имеющим RGB – значения, равные  $(255p_1, 255p_2, 0)$ .
- Оверлейный режим: Цвета точек класса «дорога» остается не изменяться. Для заливки фона каждая точка фона с исходными RGB – значениями  $(r, g, b)$  преобразуется в точку с RGB – значениями, например,  $(220, g / 3, b / 3)$ .

Результаты распознавания по шагам повторения итераций показаны ниже в табл. 6.1.

Таблица 6.1. Результаты распознавания

Итерация	Результат распознавания	
	Обычный режим	Смешанный режим
Независимое обучение		
1-ая		
2-ая		

Итерация	Результат распознавания	
	Обычный режим	Смешанный режим
3-ья		
4-ая		
5-ая		
6-ая		
7-ая		
8-ая		

Итерация	Результат распознавания	
	Обычный режим	Смешанный режим
9-ая		
10-ая		

Из таблицы 6.1 видно, что после независимого обучения существуют многие точки, которые нечетко распределены к классам. В смешанном режиме они отображены смешанными по вероятностям цветами (не красным, не зеленым, а похожими на серый цвет). После первой итерации число таких точек резко уменьшается. Результат распознавания также улучшается, т.к. многие мелкие ошибочные области удаляются.

Но дальше видно, что результаты распознавания в обычном и смешанном режимах, особенно после трех первых итераций, почти одинаковы. Повторение итераций слабо меняет результат распознавания. Повторение итераций приближает апостериорные маргинальные вероятности к нулям и единицам, что приводит к неэффективности дальнейших шагов процедуры комбинирования ациклических графов.

Для решения данной проблемы предложим схему распознавания изображений с обновлением апостериорных маргинальных вероятностей, которая заключается в следующем:

- 1) Сформируем данные учителя и выполним независимое обучение.
- 2) Выполним итерации до тех пор, пока апостериорные маргинальные вероятности не перестанут меняться.

3) Обновим апостериорные маргинальные вероятности:

Пусть имеется  $m$  классов. Точка  $t \in T$  имеет вероятности принадлежности к классам  $(p_t^1, p_t^2, \dots, p_t^m)$ , где  $p_t^i = p_t(x_t = i | Y)$ ,  $i = 1, \dots, m$ .

Пусть  $i^* = \arg \max_{i=1, \dots, m} p_t^i$ ,  $p_t^{i^*} = \max_{i=1, \dots, m} p_t^i$ ,  $t \in T$ .

$\rho$  – коэффициент обновления ( $0 < \rho < 1$ ).

Тогда апостериорные маргинальные вероятности точки  $t \in T$  обновляются по следующим формулам:

$$\begin{aligned} \pi_t^{i^*} &= \rho p_t^{i^*}, \\ \pi_t^j &= p_t^j (1 - \rho p_t^{i^*}) / \sum_{k=1, k \neq i^*}^m p_t^k, j = 1, \dots, m, j \neq i^*. \end{aligned} \quad (6.1)$$

4) Выполним итерации до тех, пор пока апостериорные маргинальные вероятности не перестанут меняться.

5) Повторим шаги 3) и 4) до тех пор, пока число ошибок после шага 4) не перестанет меняться.

В данной процедуре мы пытаемся обновлять апостериорные маргинальные вероятности так, чтобы результат распознавания не изменился, т.е. если точка  $t \in T$  принадлежит  $i$ -ому классу, то после обновления она также принадлежит тому же классу. Для этого необходимо выполнить условие:

$$\pi_t^{i^*} > \pi_t^j, j = 1, \dots, m, j \neq i^*, \text{ т.е.}$$

$$\rho p_t^{i^*} > p_t^j (1 - \rho p_t^{i^*}) / \sum_{k=1, k \neq i^*}^m p_t^k, j = 1, \dots, m, j \neq i^*.$$

Очевидно, что:

$$\sup_{p_t^k, k=1, \dots, m, k \neq i^*} \pi_t^k = 1 - \rho p_t^{i^*}.$$

Поэтому следующее условие должно быть выполнено:

$$\rho p_t^{i^*} > 1 - \rho p_t^{i^*},$$

$$\text{или } \rho p_t^{i^*} > 1/2.$$

В общем случае:  $p_t^{i^*} \geq 1/m$ .

Таким образом, для гарантии данной цели мы должны выбрать  $\rho$  так, чтобы:  $\rho/m > 1/2$ , т.е.  $\rho > 1$ . Но это невозможно, так как  $\rho$  должно меньше 1.

Для решения этого противоречия мы установим порог обновления  $\gamma$  и обновим апостериорные маргинальные вероятности только у тех точек, для которых  $p_i^{i^*} \geq \gamma$ . Тогда  $\rho$  выбирается так, чтобы  $\rho\gamma > 1/2$ , или  $\rho > 1/2\gamma$ . Например,  $\gamma = 0.75, \rho = 0.75$ .

Очевидно, что  $\gamma \leq 1$ , т.к. если  $\gamma > 1$ , то не будет точек для обновлений.

С другой стороны  $\rho < 1$ , поэтому  $1 > 1/2\gamma$ , т.е.  $\gamma > 1/2$ .

Итак, окончательная схема распознавания изображений с обновлением апостериорных маргинальных вероятностей имеет следующий вид:

- 1) Сформируем данные учителя и выполним независимое обучение.
- 2) Выполним итерации до тех пор, пока апостериорные маргинальные вероятности не перестанут меняться.
- 3) Обновим апостериорные маргинальные вероятности:

$\gamma$  – порог обновления ( $1/2 < \gamma \leq 1$ ).

$\rho$  – коэффициент обновления ( $1/2\gamma < \rho < 1$ ).

Пусть имеется  $m$  классов. Точка  $t \in T$  имеет вероятности принадлежности к классам  $(p_t^1, p_t^2, \dots, p_t^m)$ , где  $p_t^i = p_t(x_t = i | Y)$ ,  $i = 1, \dots, m$ .

Пусть  $i^* = \arg \max_{i=1, \dots, m} p_t^i, p_t^{i^*} = p_t^{i^*} = \max_{i=1, \dots, m} p_t^i$ .

Если  $p_t^{i^*} \geq \gamma$ , то апостериорные маргинальные вероятности точки  $t$  обновляются по следующим формулам:

$$\pi_t^{i^*} = \rho p_t^{i^*},$$

$$\pi_t^j = p_t^j (1 - \rho p_t^{i^*}) / \sum_{k=1, k \neq i^*}^m p_t^k, j = 1, \dots, m, j \neq i^*.$$

- 4) Выполним итерации до тех пор, пока апостериорные маргинальные вероятности не перестанут меняться.


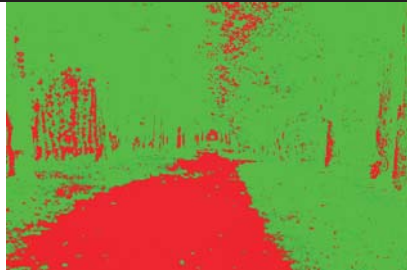
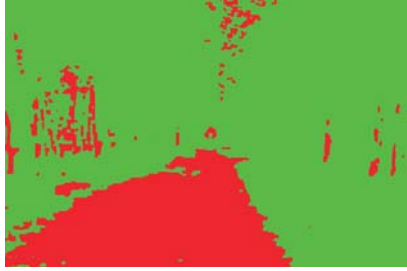
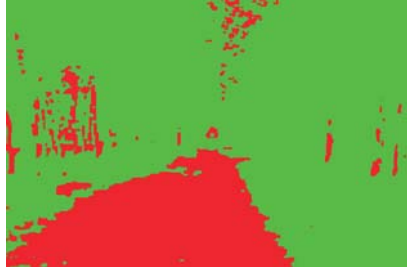


5) Повторим шаги 3) и 4) до тех пор, пока число ошибок после шага 4) не перестанет меняться.













Эксперименты показывают, что после обновления вероятностей и выполнения двух-трех итераций, апостериорные маргинальные вероятности опять быстро приближаются к нулям и единицам. Поэтому для ускорения скорости обработки в данной схеме после каждого обновления апостериорных вероятностей (шаг 3) предложим выполнить две итерации (шаг 4).

Попробуем распознать старое изображение (рис. 6.8а) по новой схеме распознавания. Результаты распознавания по шагам показаны в табл. 6.2.

Таким образом, с помощью новой схемы распознавания изображений с обновлением апостериорных маргинальных вероятностей мы достигли более хорошего результата распознавания, благодаря удалению многих мелких ошибочных областей.

Таблица 6.2. Результаты распознавания

Итерация	Результат распознавания	
	Обычный режим	Смешанный режим
Независимое обучение		
10-ая итерация без обновления		
10-ая с обновлением		

Итерация	Результат распознавания	
	Обычный режим	Смешанный режим
20-ая с обновлением		
30-ая с обновлением		
40-ая с обновлением		
50-ая с обновлением		
100-ая с обновлением		
200-ая с обновлением		

Результат распознавания в оверлейном режиме приведен ниже на рис. 6.15.

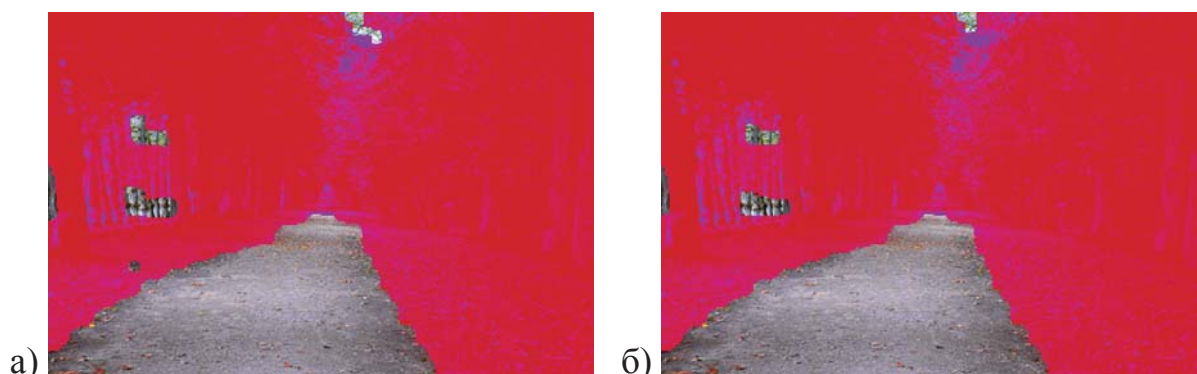


Рисунок 6.15 – Результаты распознавания изображения «Дорога» в оверлейном режиме ( $\gamma = 0.75, \rho = 0.75$ , две итерации после обновления):

а) 100 обновлений; б) 200 обновлений

Видно, что в результате распознавания (рис. 6.15) еще остаются ошибочные области, которые на самом деле не принадлежат классу «дорога».

Попробуем поменять коэффициент обновления  $\rho$  и порок  $\gamma$ , чтобы удалить эти ошибочные области. Наша цель – сделать данные более хаотическим. Поэтому попытаем уменьшить коэффициент обновления  $\rho$ . Пусть возьмем  $\gamma = 0.85$  и  $\rho = 0.6$ . Новый результат распознавания приведен на рис. 6.16.

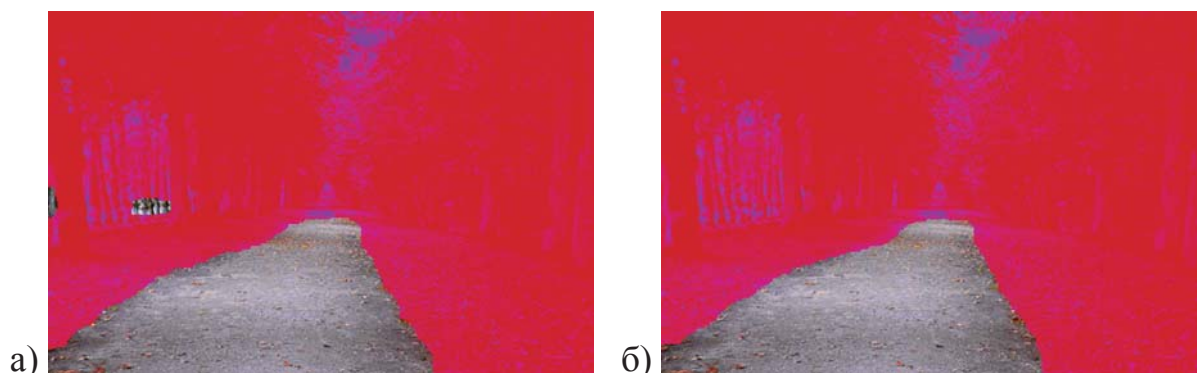


Рисунок 6.16 – Результаты распознавания изображения «Дорога» в оверлейном режиме ( $\gamma = 0.85, \rho = 0.6$ , две итерации после обновления):

а) 100 обновлений; б) 200 обновлений

Из рис. 6.16 видно, с помощью новых значений коэффициента и порога обновления  $\gamma = 0.85, \rho = 0.6$  нам удалось удалить все ошибочные области.



Алгоритм четко выделил объект «дорога» из фона. Кроме этого, важно отметить, что при новых значениях параметров граница между объектом и фоном становится более гладкой (рис. 6.16б), а не ухабистой как на рис. 6.15.

Попробуем ускорить скорость обработки уменьшением числа итераций после каждого обновления вероятностей. Раньше мы предлагали выполнять две итерации после каждого обновления. Теперь рассмотрим, как будет изменять результат распознавания, если после каждого обновления мы выполним только одну итерацию. Новый результат распознавания приведен на рис. 6.17.



Рисунок 6.17 – Результат распознавания изображения «Дорога» в оверлейном режиме ( $\gamma = 0.85, \rho = 0.6$ , одна итерации после обновления) после 300 обновлений

Из рис. 6.17 видно, что при значениях параметров  $\gamma = 0.85, \rho = 0.6$  уменьшение числа итераций после обновления вероятностей приводит к более плохому результату, в котором ошибочные области не удаляются, а граница между двумя классами стремится к перпендикулярной форме. Это объясняется тем, что выполнение одной итерации после обновления вероятностей не успевает сделать апостериорные вероятности приближенными к единицам и нулям. Причем значение порога обновления большое  $\gamma = 0.85$ . Поэтому вероятности во многих точках не обновляются, особенно в тех точках, которые близки к границе между классами. Вследствие этого обновление вероятностей становится неровным, что в свою очередь приводит к плохому результату распознавания.

Попробуем уменьшить порог обновления, сохраняя значение коэффициента обновления. Например,  $\gamma = 0.75, \rho = 0.6$ . Тогда  $\gamma\rho = 0.45 < 0.5$ , и поэтому обновление апостериорных вероятностей может изменять текущий результат распознавания, т.е. могут существовать такие точки (те точки, у которых  $Prob_{max} < 0.5 / 0.6 = 0.8333$ ), которые до обновления принадлежат одному классу, а после обновления другому. Будем называть эти точки «рискованными». Очевидно, если число рискованных точек большое, то обновление вероятностей будет полностью изменять текстурно-структурную характеристику изображения, что приводит к такому хаотическому результату. Мы с этим риском согласимся и рассмотрим, как алгоритм распознавания будет работать. Новый результат распознавания изображения «Дорога» приведен ниже на рис. 6.18.



Рисунок 6.18 – Результат распознавания изображения «Дорога» в обычном режиме ( $\gamma = 0.75, \rho = 0.6$ , одна итерации после обновления) после 10 обновлений

Из рис. 6.18 видно, что уменьшение порога обновления приводит к хаотическому результату распознавания, который фактически является следствием появления многих рискованных точек.

Для уменьшения числа рискованных точек мы можем увеличить значение коэффициента обновления. Например,  $\rho = 0.65$  или  $\rho = 0.7$ . Тогда рискованными являются только те точки, у которых  $Prob_{max} < 0.5 / 0.65 = 0.769$  или  $Prob_{max} < 0.5 / 7 = 0.714$ . Действительно, число таких точек будет меньше, чем в прошлом случае, когда  $\rho = 0.6$ . В свою очередь, значение порога об-

новления также может еще уменьшаться. Например,  $\gamma = 0.7$  или  $\gamma = 0.65$ . Соответствующие разным парам параметров обновления результаты распознавания показаны на рис. 6.19.

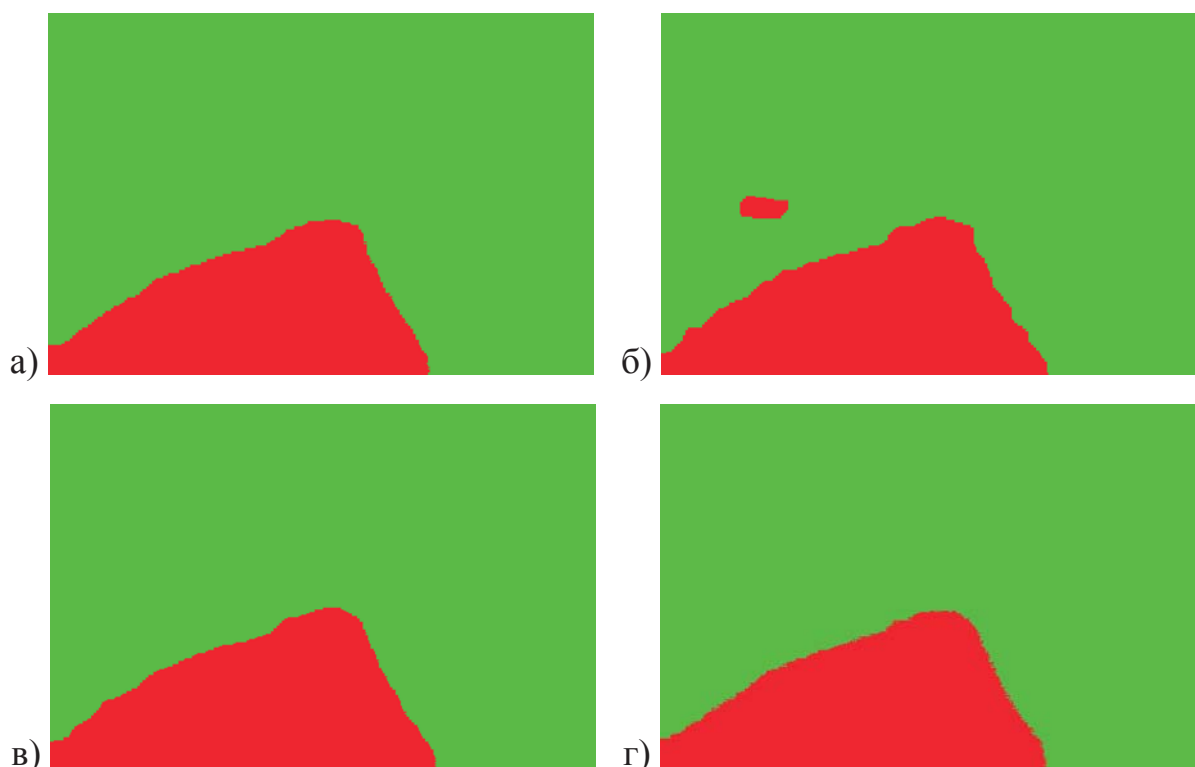


Рисунок 6.19 – Результаты распознавания изображения «Дорога» в оверлейном режиме (одна итерация после обновления):

а)  $\gamma = 0.75, \rho = 0.65$ , 200 обновлений; б)  $\gamma = 0.75, \rho = 0.7$ , 350 обновлений; в)  $\gamma = 0.7, \rho = 0.7$ , 220 обновлений ; г)  $\gamma = 0.65, \rho = 0.65$ , 40 обновлений

Из рис. 6.19 видно, что пара значений параметров  $\gamma = 0.75, \rho = 0.7$  не может удалить все ошибочные области (рис. 6.19б), даже с большим числом обновления вероятностей. Все остальные пары значений дают почти одинаковые результаты. Отличие между ними заключается в виде границы и скорости. Пара значений  $\gamma = 0.65, \rho = 0.65$  быстро удаляет все ошибочные области и дает окончательный результат. Однако граница между классами нечеткая и шероховатая. Пара значений  $\gamma = 0.7, \rho = 0.7$  требует чуть больше времени выполнения, но дает более гладкую границу, чем пара значений  $\gamma = 0.75, \rho = 0.65$ .

Рассмотрим другое изображение «мангровый лес вдоль реки Шабанг в Малайзии» (рис. 6.20а [60]). Сравним результаты его обработки при двух наборах параметров  $\gamma = 0.85, \rho = 0.6$ , две итерации после обновления; и  $\gamma = 0.7, \rho = 0.7$ , одна итерация после обновления.

Предполагается, что необходимо разделить изображение на три области: река, лес и небо. Т.е. речь идет о задаче трехклассовой сегментации изображения. Пусть данные учителя сформулированы как на рис. 6.20б.

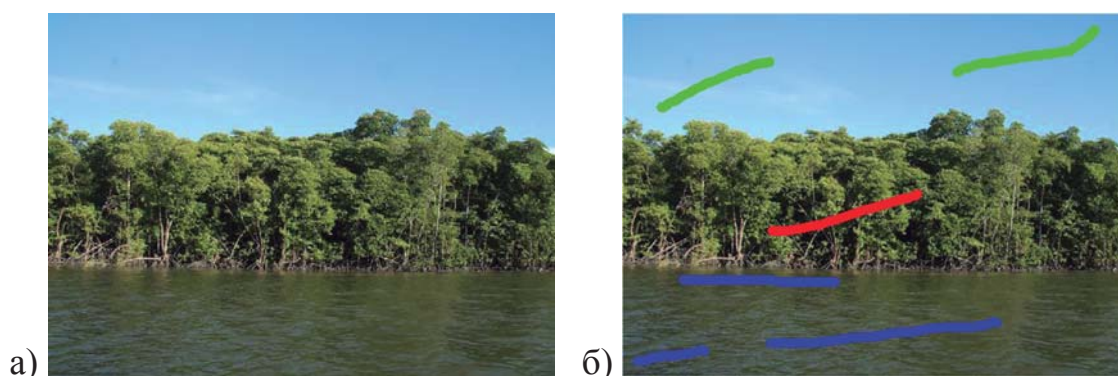


Рисунок 6.20 – Изображение «мангровый лес вдоль реки Шабанг в Малайзии»: а) исходное изображение; б) данные учителя

Результаты независимого обучения и распознавания без обновления апостериорных вероятностей показаны на рис. 6.21.

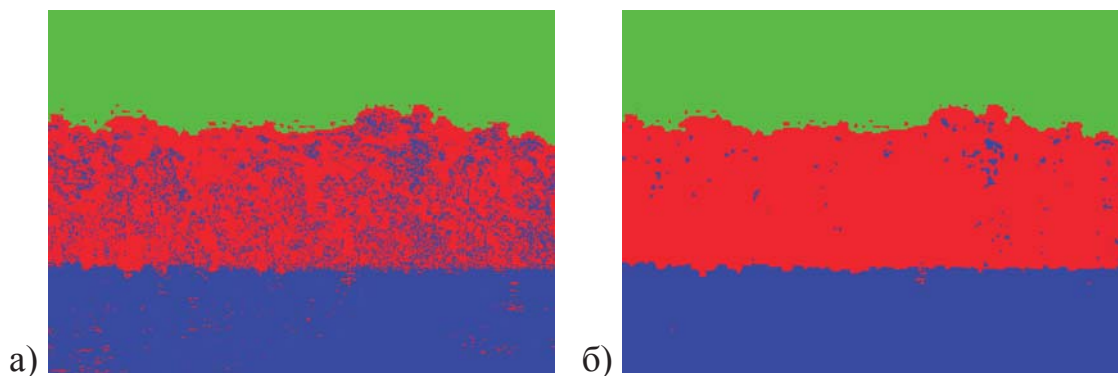


Рисунок 6.21 – Результаты распознавания: а) независимое обучение; б) распознавание без обновления после 10 итераций

Результаты распознавания с обновлением вероятностей при наборе параметров  $\gamma = 0.85, \rho = 0.6$ , две итерации после обновления приведены на рис. 6.22.

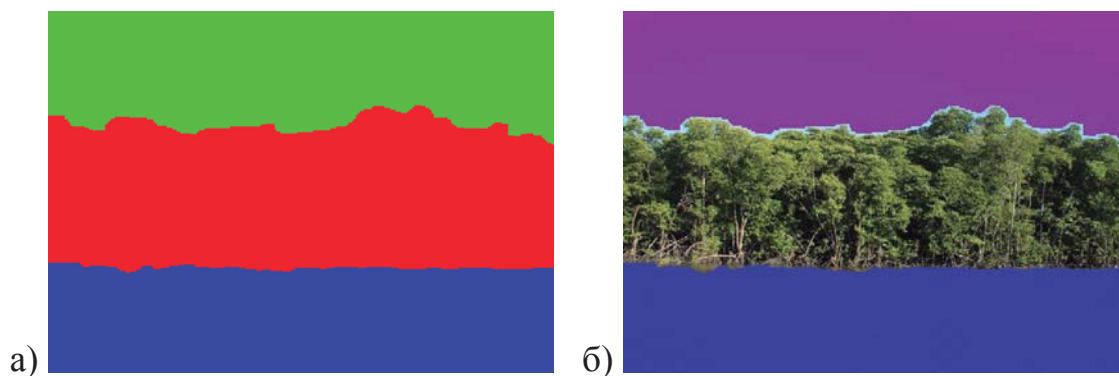


Рисунок 6.22 – Результаты распознавания с обновлением ( $\gamma = 0.85$ ,  $\rho = 0.6$ , две итерации после обновления): а) обычный режим; б) оверлейный режим (20 обновлений)

Результаты распознавания с обновлением вероятностей при наборе параметров  $\gamma = 0.7, \rho = 0.7$ , одна итерация после обновления приведены на рис. 6.23.

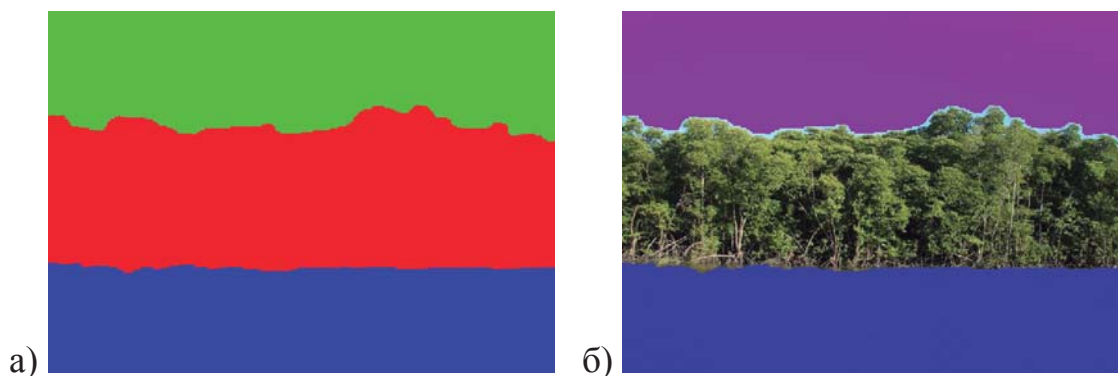


Рисунок 6.23 – Результаты распознавания с обновлением ( $\gamma = 0.7$ ,  $\rho = 0.7$ , одна итерация после обновления): а) обычный режим; б) оверлейный режим (15 обновлений)

Из рис. 6.22–6.23 видно, что два набора параметров дают почти одинаковые результаты распознавания. Причем набор параметров  $\gamma = 0.7, \rho = 0.7$ , одна итерация после обновления работает быстрее примерно в три раза ( $20 \times 2 / 15 \approx 2.67$ ), чем набор параметров  $\gamma = 0.85, \rho = 0.6$ , две итерации после обновления.

Теперь пусть мы хотим отделить только лес от фона. Данные учителя сформулированы как на рис. 6.24.

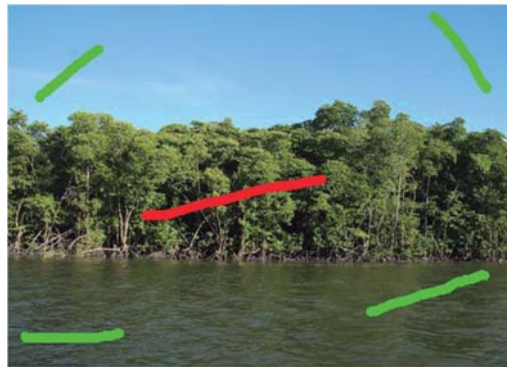


Рисунок 6.24 – Выбор данных учителя для выделения леса

Результаты независимого обучения и распознавания без обновления апостериорных вероятностей показаны на рис. 6.25.

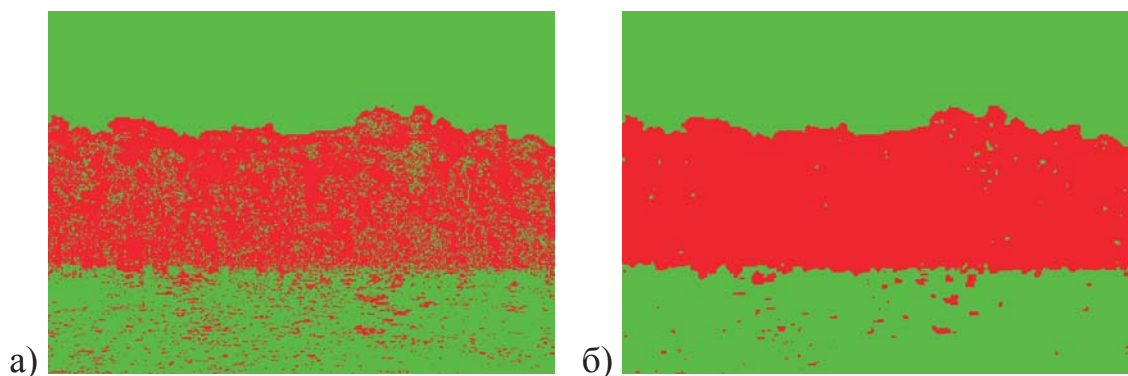


Рисунок 6.25 – Результаты распознавания: а) независимое обучение; б) распознавание без обновления после 10 итераций

Результаты распознавания с обновлением вероятностей при наборе параметров  $\gamma = 0.85, \rho = 0.6$ , две итерации после обновления приведены на рис. 6.26.

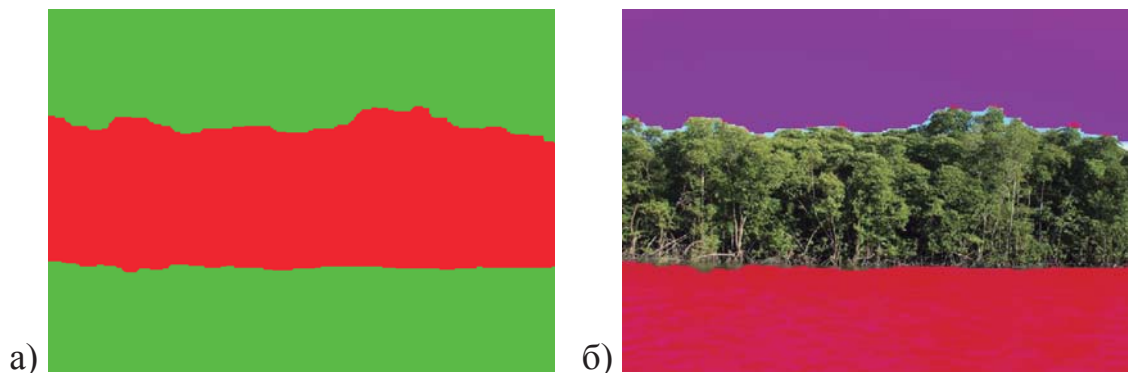


Рисунок 6.26 – Результаты распознавания с обновлением ( $\gamma = 0.85, \rho = 0.6$ , две итерации после обновления): а) обычный режим; б) оверлейный режим (40 обновлений)

Результаты распознавания с обновлением вероятностей при наборе параметров  $\gamma = 0.7, \rho = 0.7$ , одна итерация после обновления приведены на рис. 6.27.

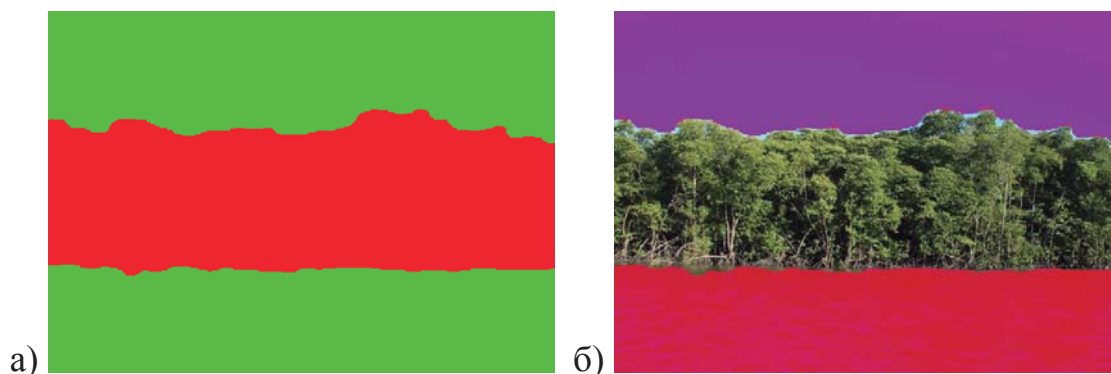


Рисунок 6.27 – Результаты распознавания с обновлением ( $\gamma = 0.7, \rho = 0.7$ , одна итерация после обновления): а) обычный режим; б) оверлейный режим (50 обновлений)

Из рис. 6.26–6.27 видно, что два набора параметров опять дают почти одинаковые результаты распознавания. Причем набор параметров  $\gamma = 0.7, \rho = 0.7$ , одна итерация после обновления работает быстрее примерно в полтора раза ( $40 \times 2 / 50 \approx 1.6$ ), чем набор параметров  $\gamma = 0.85, \rho = 0.6$ , две итерации после обновления.

Дальше попробуем отделить реку от фона. Пусть данные учителя сформулированы как на рис. 6.28.



Рисунок 6.28 – Выбор данных учителя для выделения реки

Результаты независимого обучения и распознавания без обновления апостериорных вероятностей показаны на рис. 6.29.

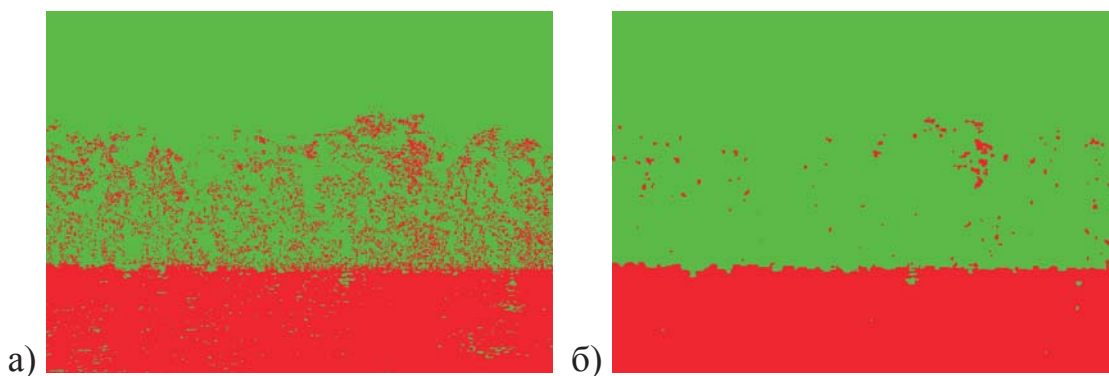


Рисунок 6.29 – Результаты распознавания: а) независимое обучение; б) распознавание без обновления после 10 итераций

Результаты распознавания с обновлением вероятностей при наборе параметров  $\gamma = 0.85, \rho = 0.6$ , две итерации после обновления приведены на рис. 6.30.

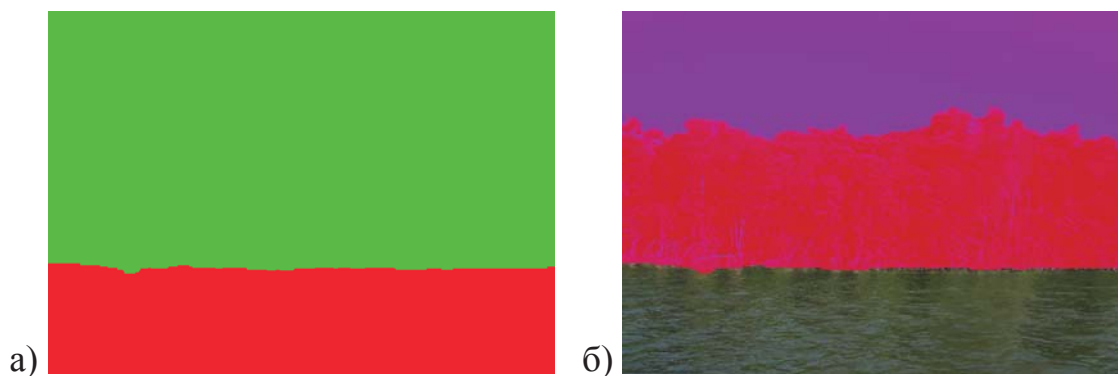


Рисунок 6.30 – Результаты распознавания с обновлением ( $\gamma = 0.85, \rho = 0.6$ , две итерации после обновления): а) обычный режим; б) оверлейный режим (10 обновлений)

Результаты распознавания с обновлением вероятностей при наборе параметров  $\gamma = 0.7, \rho = 0.7$ , одна итерация после обновления приведены на рис. 6.31.

Снова мы получаем почти одинаковые результаты распознавания (рис. 6.30–6.31). Причем набор параметров  $\gamma = 0.7, \rho = 0.7$ , одна итерация после обновления работает немного быстрее ( $10 \times 2 / 15 \approx 1.33$ ), чем набор параметров  $\gamma = 0.85, \rho = 0.6$ , две итерации после обновления.



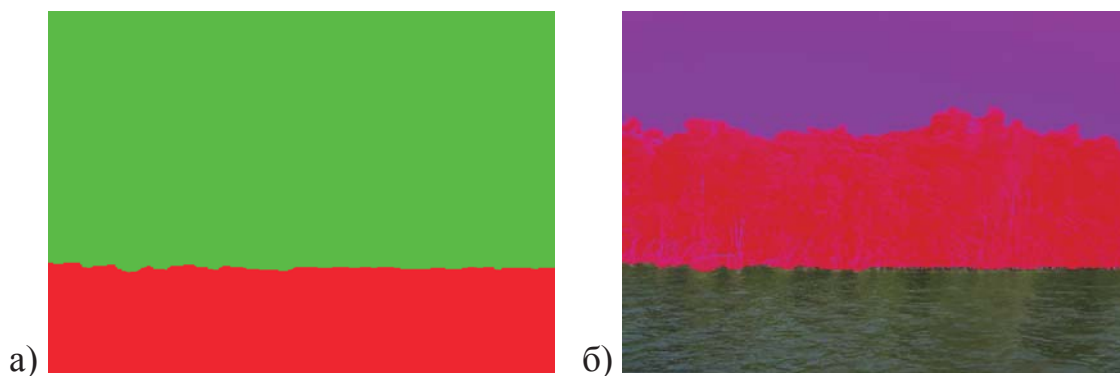


Рисунок 6.31 – Результаты распознавания с обновлением ( $\gamma = 0.7$ ,  $\rho = 0.7$ , одна итерация после обновления): а) обычный режим; б) оверлейный режим (15 обновлений)

Дальше попробуем отделить небо от фона. Пусть данные учителя сформулированы как на рис. 6.32.



Рисунок 6.32 – Выбор данных учителя для выделения неба

Результаты независимого обучения и распознавания без обновления апостериорных вероятностей показаны на рис. 6.33.

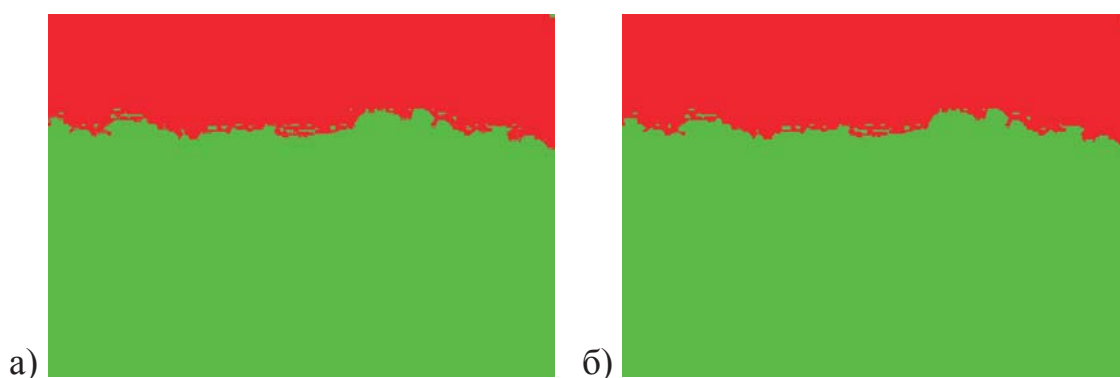


Рисунок 6.33 – Результаты распознавания: а) независимое обучение; б) распознавание без обновления после 10 итераций

Результаты распознавания с обновлением вероятностей при наборе параметров  $\gamma = 0.85, \rho = 0.6$ , две итерации после обновления приведены на рис. 6.34.

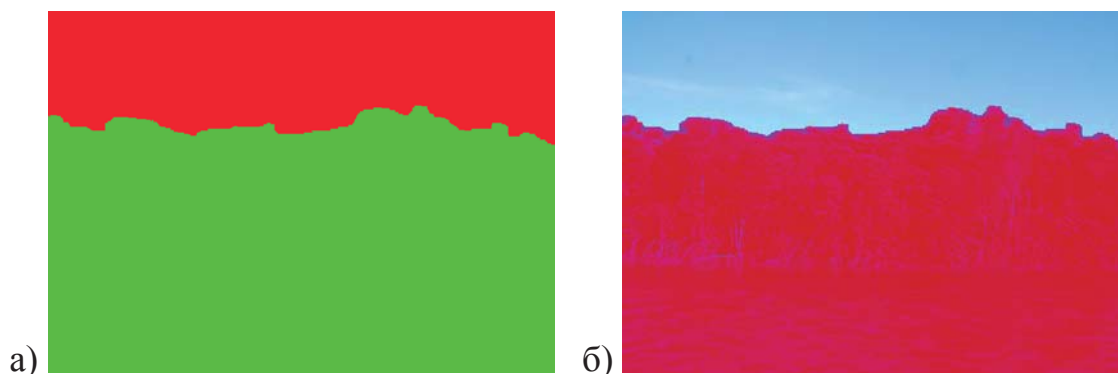


Рисунок 6.34 – Результаты распознавания с обновлением ( $\gamma = 0.85, \rho = 0.6$ , две итерации после обновления): а) обычный режим; б) оверлейный режим (10 обновлений)

Результаты распознавания с обновлением вероятностей при наборе параметров  $\gamma = 0.7, \rho = 0.7$ , одна итерация после обновления приведены на рис. 6.35.

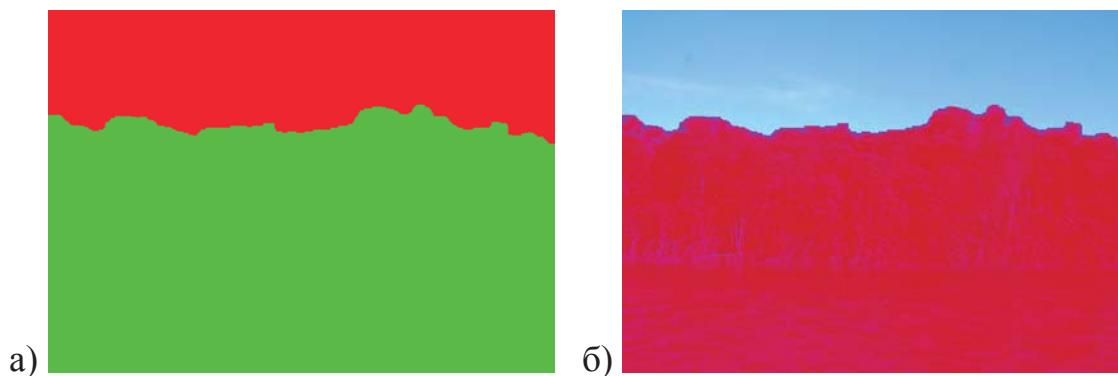


Рисунок 6.35 – Результаты распознавания с обновлением ( $\gamma = 0.7, \rho = 0.7$ , одна итерация после обновления): а) обычный режим; б) оверлейный режим (11 обновлений)

Результаты распознавания на рис. 6.34–6.35 еще раз подтверждают, что два набора параметров дают почти одинаковые результаты, но набор параметров  $\gamma = 0.7, \rho = 0.7$ , одна итерация после обновления работает быстрее, чем набор параметров  $\gamma = 0.85, \rho = 0.6$ , две итерации после обновления.

Таким образом, для получения хорошего результата распознавания с быстрой скоростью обработки мы можем использовать набор параметров обновления:  $\gamma = 0.7, \rho = 0.7$ , одна итерация после обновления.

### **6.5 Обработка изображений известных баз**

Известный алгоритм TRWS был разработан коллективом Миддлберийского колледжа в США. На их официальном сайте были использованы 3 изображения для иллюстрации работы алгоритма TRWS [57]. Попробуем распознать эти изображения предложенными алгоритмами.

Первое изображение – цветок (рис. 6.8а). Пусть данные учителя выбираются, как показано на рис. 6.8б. Результат распознавания этого изображения приведен на рис. 6.36.

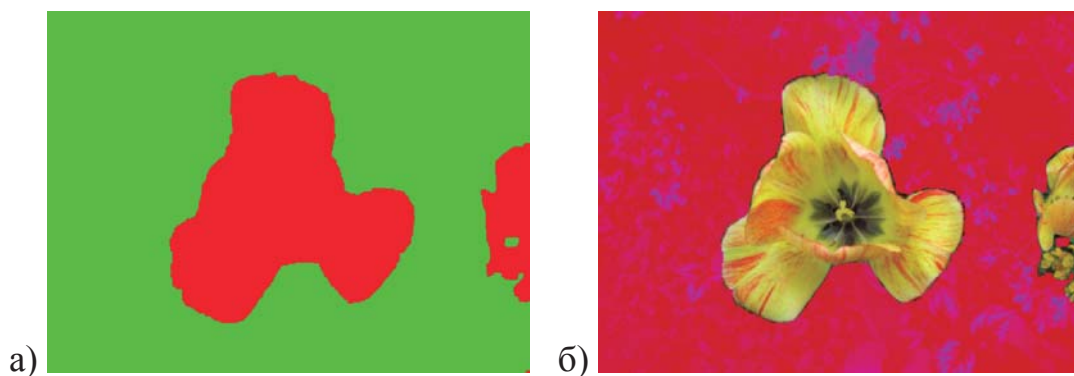


Рисунок 6.36 – Результаты распознавания с обновлением: а) обычный режим; б) оверлейный режим

Второе изображение – камень (рис. 6.37б). Пусть данные учителя выбираются, как показано на рис. 6.37б. Результат распознавания этого изображения приведен на рис. 6.37в–г.

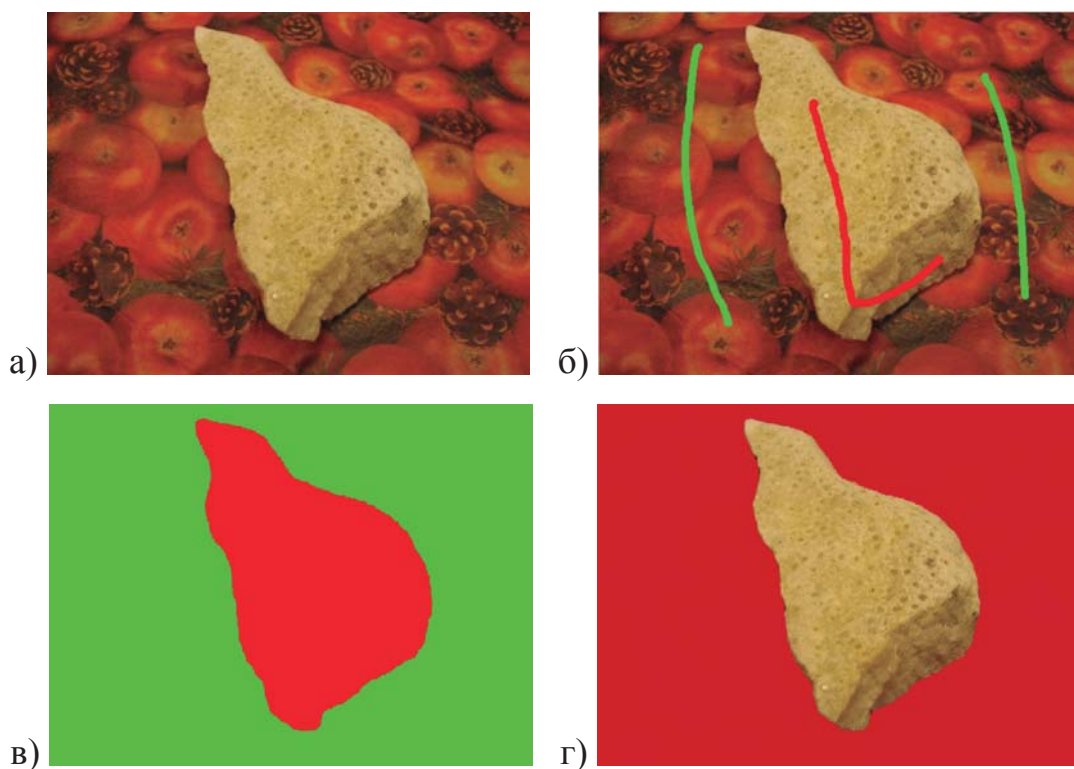


Рисунок 6.37 – Изображение «камень»: а) исходное изображение; б) выбор данных учителя; в) результат в обычном режиме; г) результат в оверлейном режиме

Третье изображение – человек (рис. 6.38б). Пусть данные учителя выбираются, как показано на рис. 6.38б. Результат распознавания этого изображения приведен на рис. 6.38в–г.

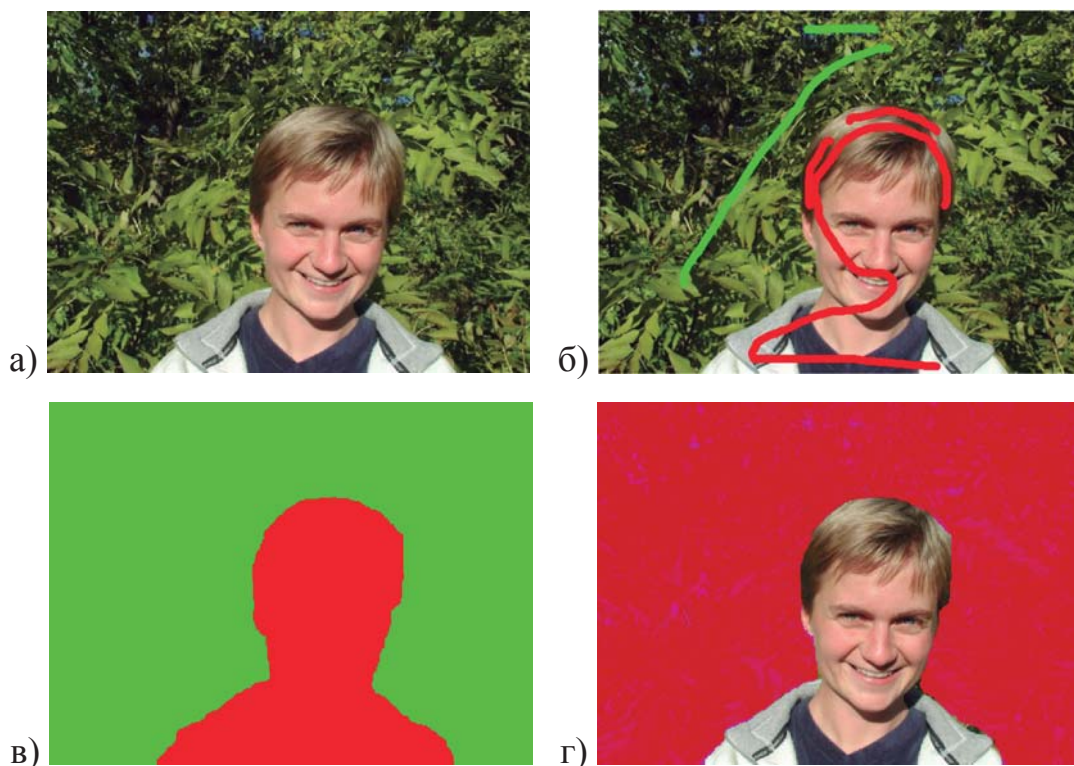


Рисунок 6.38 – Изображение «человек»: а) исходное изображение; б) выбор данных учителя; в) результат в обычном режиме; г) результат в оверлейном режиме

Другой известной является база изображений BSDS500 [59]. На рис. 6.39–40 приведены примеры обработки ее изображений.

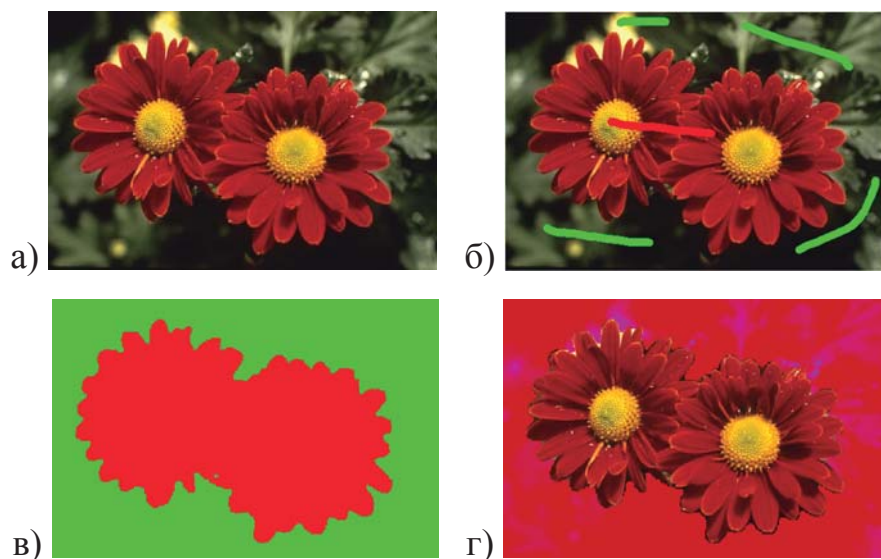


Рисунок 6.39 – Изображение «маргаритка»: а) исходное изображение; б) выбор данных учителя; в) результат в обычном режиме; г) результат в оверлейном режиме

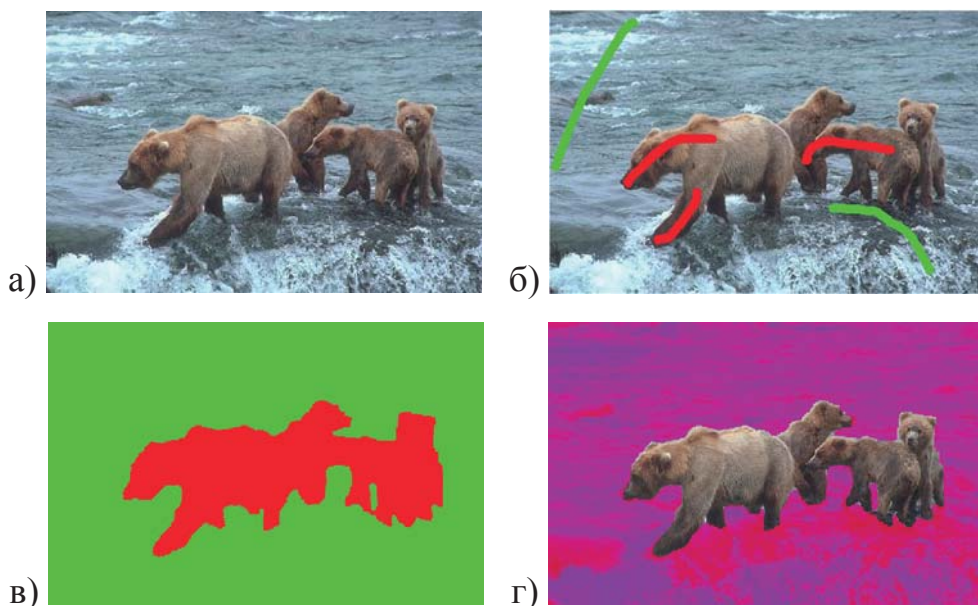


Рисунок 6.40 – Изображение «медведи»: а) исходное изображение; б) выбор данных учителя; в) результат в обычном режиме; г) результат в оверлейном режиме

Примеры обработки спутниковых изображений в базе данных [55] компании *FPT Software* приведены на рис. 6.41–42.

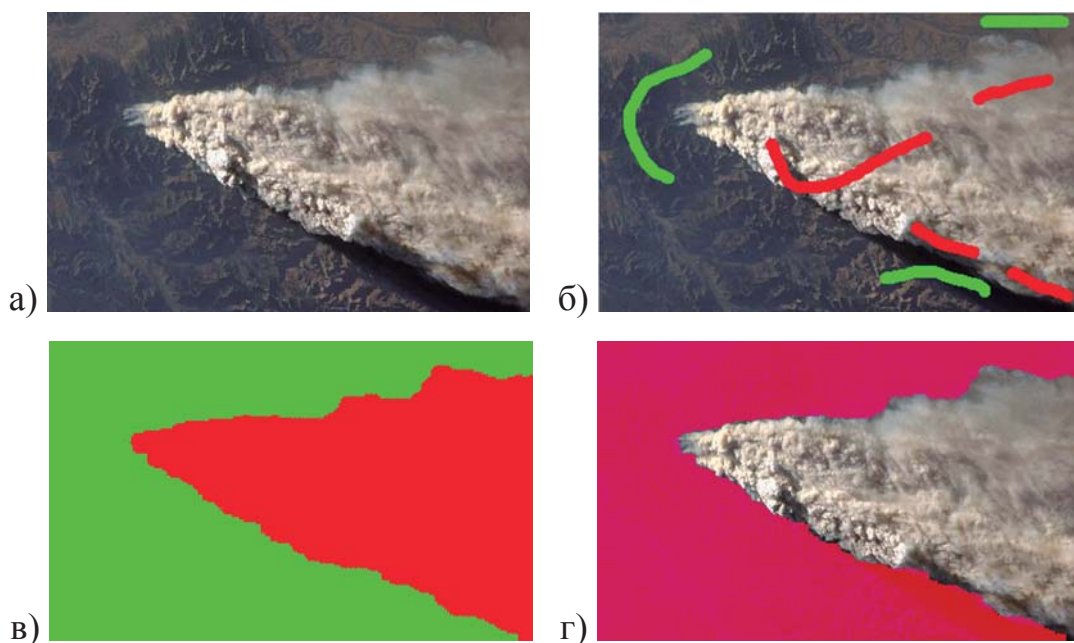


Рисунок 6.41 – Изображение «дым»: а) исходное изображение; б) выбор данных учителя; в) результат в обычном режиме; г) результат в оверлейном режиме

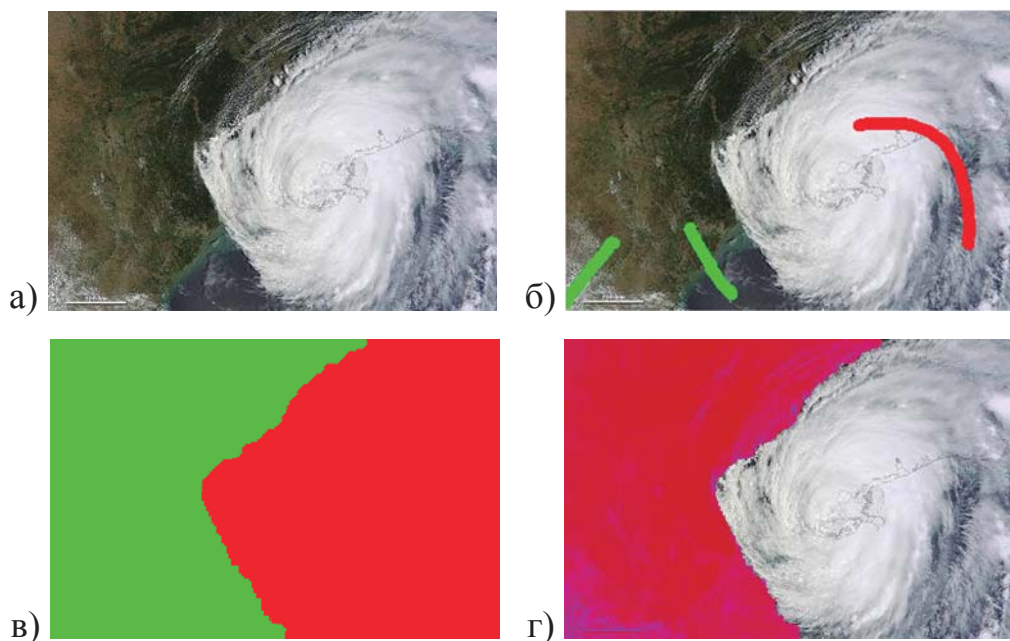


Рисунок 6.42 – Изображение «глаз бури»: а) исходное изображение; б) выбор данных учителя; в) результат в обычном режиме; г) результат в оверлейном режиме

Примеры обработки изображений в базе данных [56] компании *FTS Corp* приведены на рис. 6.43–44.

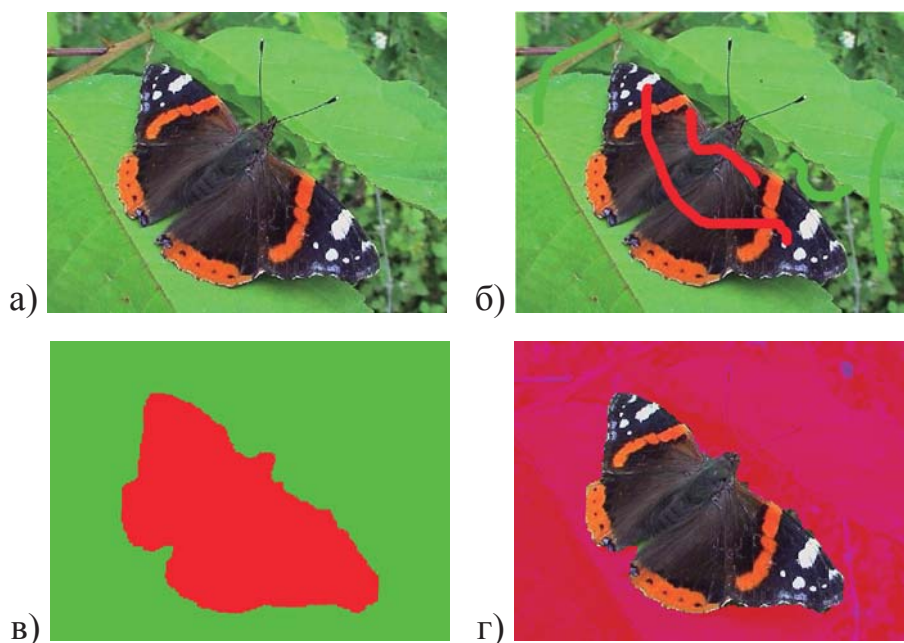


Рисунок 6.43 – Изображение «бабочка»: а) исходное изображение; б) выбор данных учителя; в) результат в обычном режиме; г) результат в оверлейном режиме

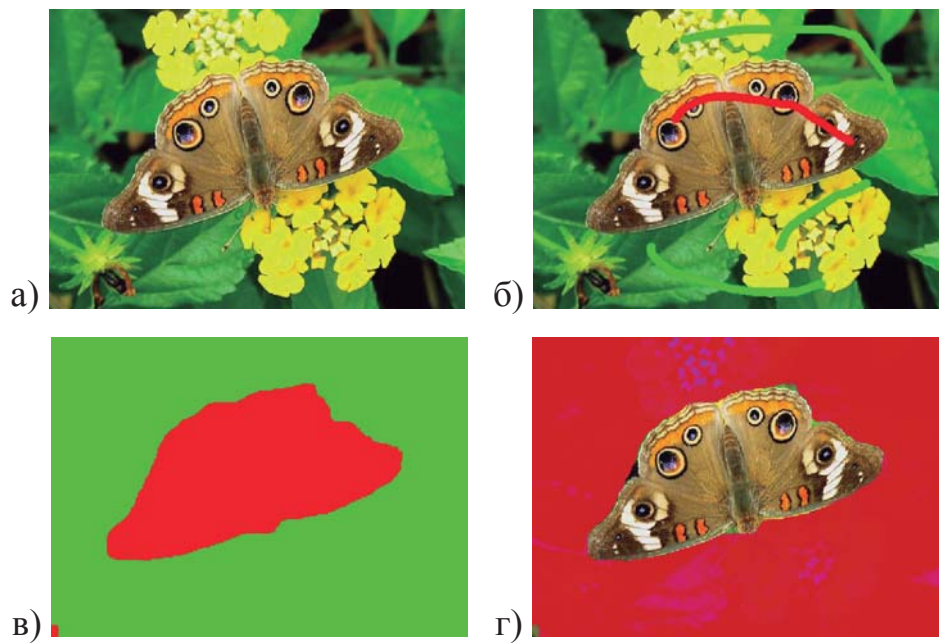


Рисунок 6.44 – Изображение «бабочка»: а) исходное изображение; б) выбор данных учителя; в) результат в обычном режиме; г) результат в оверлейном режиме



## ЗАКЛЮЧЕНИЕ

В данной работе рассмотрена задача распознавания образов в массивах взаимосвязанных данных и, в частности, задача обработки текстурных изображений. Марковские модели, управляющие сменой скрытых классов распознаваемых объектов, оказались эффективны при обработке линейно упорядоченных массивов с цепочечным соседством их элементов. Однако довольно часто на практике графы соседства имеют произвольный вид и содержат циклы. В таких случаях задача распознавания марковских случайных полей является весьма трудоемкой и обладает свойствами задачи класса  $NP$ . Однако существуют частные виды данной задачи, допускающие ее эффективное решение. В частности, ранее был предложен эффективный базовый алгоритм распознавания по ациклическому графу соседства. Ранее также было предложено комбинировать ациклические графы соседства, чтобы уменьшить потери, связанные с древовидной аппроксимацией исходного графа соседства. Однако в таких алгоритмах значение марковского параметра ранее задавалось эвристически без поиска его оптимального значения диагонального элемента.

В данной работе были доказаны вырожденные свойства базового алгоритма распознавания при предельных значениях диагонального элемента. Были сформулированы постановки задачи подбора диагонального элемента для заданного ациклического графа и задачи подбора параметров комбинирования ациклических графов соседства.

В данной работе построены алгоритмы подбора диагонального элемента для заданного ациклического графа соседства на основе принципа максимизации апостериорной совместной вероятности скрытых классов. Также разработаны алгоритмы подбора параметров комбинирования ациклических графов соседства по методу покоординатного спуска Гаусса-Зайделя. Суть этих алгоритмов – это включение поиска значения диагонального элемента в схему Гаусса-Зайделя при подборе весов графов. До этого были проведены предварительные эксперименты для оценки характеристики изменения числа

ошибок при совместном варьировании весов графов и значения диагонального элемента в случае как однократного, так и многократного распознавания. На основе предварительных экспериментов были сделаны выводы о необходимости использования многократного распознавания в новом алгоритме обучения.

В данной работе проведено экспериментальное исследование новых алгоритмов. Результаты экспериментов показывают, что алгоритм подбора диагонального элемента для заданного ациклического графа на основе независимого обучения является наихудшим. Если в нем на второй итерации вместо пересчитанного по независимому обучению диагонального элемента использовать заданный, то результат улучшается. Но есть случаи, когда алгоритм подбора диагонального элемента выигрывает у базового алгоритма и, наоборот, есть случаи, когда алгоритм подбора диагонального элемента проигрывает. Алгоритм подбора диагонального элемента по схеме Гаусса-Зайделя дает лучший результат и выигрывает у остальных алгоритмов. Кроме этого, оптимальное значение диагонального элемента резко уменьшает число итераций (до одной – двух) базового алгоритма для любого ациклического графа соседства.

Результаты экспериментов также показывают, что разработанные в данной работе алгоритмы подбора параметров комбинирования ациклических графов соседства позволяют резко улучшить качество распознавания по сравнению с исходным алгоритмом подбора параметров по схеме Гаусса-Зайделя, в котором элемент  $q$  не подбирается и задается эвристически. Разработанные алгоритмы подбора параметров комбинирования графов соседства обеспечивают высокое качество распознавания и, в частности, оказываются сравнимыми по качеству распознавания с алгоритмом TRWS, который сегодня считается одним из наиболее эффективных алгоритмов распознавания.

Построенные в данной работе алгоритмы являются алгоритмами того же класса по качеству распознавания, что и TRWS, который решает глобальную задачу минимизации гиббсовской энергии связей.

Таким образом, решая локальную задачу минимизации ошибок распознавания на основе численного оценивания апостериорных вероятностей скрытых классов в элементах массива данных, мы построили алгоритмы с высоким качеством распознавания.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Вазан М. Стохастическая аппроксимация. – М.: Мир, 1972. – 295 с.
2. Вапник В.Н., Червоненкис А.Я. Теория распознавания образов. Статистические проблемы обучения: учеб. пособие. – М.: Наука, 1974. – 416 с.
3. Васильев Ф.П. Методы оптимизации. – М.: Факториал Пресс, 2002. – 824с.
4. Ветров Д.П., Кропотов Д.А. Байесовские методы машинного обучения: учеб. пособие. – М.: МГУ, 2007. – 132 с.
5. Гонсалес Р., Вудс Р. Цифровая обработка изображений / пер. с англ. – М.: Техносфера, 2006, 1072 с.
6. Гмурман В.Е. Теория вероятностей и математическая статистика: учеб. пособие. – 12-ое изд. – М.: Высш. Обр., 2007, 478 с.
7. Двоенко С.Д. Алгоритмы распознавания взаимосвязанных объектов: дис. ... док. физ-мат. наук. – Тула: Тульский гос. ун-т, 2001. – 200 с.
8. Двоенко С.Д., Копылов А.В., Моттль В.В. Задача распознавания образов в массивах взаимосвязанных объектов. Постановка задачи и основные предположения // Автоматика и телемеханика. – 2004. – № 1. – С. 143–158.
9. Двоенко С.Д., Копылов А.В., Моттль В.В. Задача распознавания образов в массивах взаимосвязанных объектов. Алгоритм распознавания // Автоматика и телемеханика. – 2005. – № 12. – С. 162–176.
10. Двоенко С.Д., Савенков С.Д. Древоподобные марковские модели в анализе массивов взаимосвязанных данных // Искусственный интеллект. – 2006. – № 2. – С. 384–391.
11. Двоенко С.Д., Савенков С.Д. Комбинирование ациклических графов соседства в задаче распознавания марковских случайных полей // Сб. докл. конф. ММРО–14. М.: МАКС Пресс, 2009. С. 441–444.
12. Двоенко С.Д., Савенков С.Д., Шанг Д.В. Ациклические марковские модели в анализе массивов взаимосвязанных данных // Изв. ТулГУ. Естественные науки. 2010. Вып. 2. С. 173–185.

13. Дуда Р., Харт М. Распознавание образов и анализ сцен: учеб. пособие. – 2-е изд., перераб. – М.: Мир, 1976. – 512 с.
14. Колмогоров А.Н., Фомин С.В. Элементы теории функций и функционального анализа: учеб. пособие. – М.: Наука, 1976. – 542 с.
15. Моттль В.В., Мучник И.Б. Скрытые марковские модели в структурном анализе сигналов: учеб. пособие. – 1-ое изд. – М.: ФИЗМАТЛИТ, 1999. – 352 с.
16. Ту Дж., Гонсалес Р. Принципы распознавания образов. – М.: Мир, 1978. – 414 с.
17. Фукунага К. Введение в статистическую теорию распознавания образов : учеб. пособие. – 1-ое изд. – М.: Наука, 1979. – 368 с.
18. Шлезингер М., Главач В. 10 лекций по статистическому и структурному распознаванию образов : учеб. пособие. – 2-е изд., перераб. – Киев: Наукова Думка, 2004. – 545 с.
19. Alpaydin E. Introduction to machine learning. – Cambridge: MIT, MA, 2004. – 415 p.
20. Besag J. On the statistical analysis of dirty pictures (with discussion) // Journal of the Royal Statistical Society, Series B 48. – 1986. – P. 259–302.
21. Bishop C.M. Pattern Recognition and Machine Learning – New York: Springer, 2006. 738 p.
22. Bondy J.A., Murty U.S.R. Graph theory. – New York: Springer, 2008. – 654 p.
23. Brunet F. Contributions to Parametric Image Registration and 3D Surface Reconstruction: PhD dissertation. – France: IRIT, 2010. – 234 p.
24. Chou P.B. Probabilistic network inference for cooperative high and low level vision / P. B. Chou, P. R. Cooper, M. J. Swain, C. M. Brown, and L. E. Wixson // Chellappa R. and Jain A. editors, Markov Random Fields: Theory and Applications. – Boston.: Academic Press, 1993. – P. 211–243.
25. Dawid A.P. Applications of general propagation algorithm for probabilistic expert systems // Statistics and Computing. – 1992. – P. 25–36.

26. Duda R., Hart P., Stork D. Pattern classification. – 2<sup>nd</sup> Edition. – New York: Wiley, 2001. – 654 p.
27. Dvoenko S., Savenkov D. The effective recognition procedure based on tree-like markov models // Proc. 9th PRIP. – 2007. – Vol. 1, № 2. – P. 98–100.
28. Feichtinger H., Strohmer T. Gabor analysis and algorithms // Birkhauser, 1998. – 500 p.
29. Fink G. Markov models for pattern recognition. – New York: Springer, 2008. – 248 p.
30. Forsyth D.A., Ponce J. Computer vision: A modern approach. – 2<sup>nd</sup> Edition. – Prentice Hall, 2011. – 761 p.
31. Geman S., Geman D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images // IEEE Trans. – 1984. – Vol. 6, № 6. – P. 721–741.
32. Hastie T., Tibshinari R., Friedman J. The Elements of Statistical Learning. – 2<sup>nd</sup> Edition. – New York: Springer, 2009. – 746 p.
33. Kämäräinen J.K., Kyrk V., Kälviäinen H. Local and Global Gabor Features for Object Recognition // Proc. PRIA. – 2007. – Vol. 17, № 1 – P. 93–105.
34. Kindermann R., Laurie Snell J. Markov Random Fields and their applications // Rhode Island, Contemporary Mathematics. – 1980. – Vol. 1.
35. Kolmogorov V., Zabih R. What energy functions can be minimized via graph cuts? // IEEE Trans Pattern Anal Mach Intell 26, 2004.
36. Kolmogorov V. Convergent Tree-reweighted Message Passing for Energy Minimization // IEEE Pattern Analysis and Machine Intelligence (PAMI), 28(10) :1568-1583. – October 2006.
37. Li S.Z. Markov Random Field Modeling in Computer Vision. – New York: Springer Verlag, 1995. – 350 p.
38. Mottl V.V. Pattern Recognition in Spatial Data: A New Method of Seismic Explorations for Oil and Gas in Crystalline Basement Rocks / V.V. Mottl, S.D. Dvoenko, V.B. Levyant, I.B. Muchnik // Proc. 15<sup>th</sup> ICPR'2000. Spain, Barcelona. – 2000. – Vol. 3. – P. 210–213.

39. Mottl V.V. Pattern Recognition in Interrelated Data: The Problem, Fundamental Assumptions, Recognition Algorithms / V.V. Mottl, S.D. Dvoenko, A.V.Kopylov // Proc. 17<sup>th</sup> ICPR'2004. Cambridge, England, UK. – 2004. – Vol. 1. – P. 188–191.
40. Neal R., Hinton G. A view of the EM algorithm that justifies incremental, sparse and other variants // Learning in graphical models. – 1999. – P. 355-368.
41. Pratt W. Digital image processing // 4<sup>th</sup> Edition. – New York: Wiley, 2007. – 784 p.
42. Rabiner L.R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition // Proc. IEEE, 77. – 1977. – Vol. 2. – P. 257–286.
43. Saquib S.S. ML, Bouman C.A., Sauer K. Parameter Estimation for Markov Random Fields, with Applications to Bayesian Tomography // IEEE transactions on Image Processing. – Jul 1998. – Vol. 7, №7. – P. 1029–1044.
44. Schildt H. The complete reference C++. – 4<sup>th</sup> Edition. – McGraw Hill, 2003. – 1056 p.
45. Schlesinger M., Flach B. Some solvable subclasses of structural recognition problems // Czech Pattern Recognition Workshop 2000, Tomas Svoboda (ed.), Czech Pattern Recognition Society, Praha. – February 2000. – P. 55–61.
46. Szeliski R. Comparative Study of Energy Minimization Methods for Markov Random Fields / R. Szeliski, R. Zabih, D. Scharstein and others // Proc. ECCV. – 2006.
47. Vapnik V.N. Statistical Learning Theory. – New York: Wiley, 1998. – 736 p.
48. Vapnik V.N. The Nature of Statistical Learning Theory. – New York: Springer Verlag, 1995. – 188 p.
49. Wainwright M., Jordan M. Graphical models, exponential families and variational inference // Technical Report 649, University of Berkley. – 2003.
50. Won C.S., Derin H. Unsupervised segmentation of noisy and textured images using Markov random fields // CVGIP: Graphics Model and Image Processing, 54. – 1992. – P. 308–328.

51. Wu C.H., Doerschuk P.C. Tree Approximation to Markov Random Fields // IEEE transactions on Pattern Analysis and Machine Intelligence. – April 1995. – Vol. 17, № 4. – P. 391–402.
52. Yedidia J.S., Freeman W.T., Weiss Y. Generalized belief propagation // NIPS. – 2000. – P. 689–695.
53. Image dataset Caltech-101 [Электронный ресурс]. URL: [http://www.vision.caltech.edu/Image\\_Datasets/Caltech101/Caltech101.html](http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html) (дата обращения: 20.04.2013).
54. Image dataset of NASA [Электронный ресурс]. URL: <http://visibleearth.nasa.gov> (дата обращения: 20.04.2013).
55. Image dataset of FPT Software [Электронный ресурс] / Vietnam, Hanoi, 2012. – 1 CD-ROM.
56. Image dataset of FTS Corp [Электронный ресурс] / Vietnam, Hanoi, 2012. – 1 CD-ROM.
57. Middlebury Vision image dataset [Электронный ресурс]. URL: <http://vision.middlebury.edu/MRF/results/> (дата обращения: 20.04.2013).
58. Public forum [Электронный ресурс]. URL: <http://forumserver.twoplustwo.com/41/politics/could-untouched-land-really-protected-ac-society-595764/> (дата обращения: 20.04.2013).
59. The Berkeley segmentation dataset [Электронный ресурс]. URL: <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html> (дата обращения: 20.04.2013).
60. Travel information [Электронный ресурс]. URL: [http://travel.mongabay.com/malaysia/images/borneo\\_6510.html](http://travel.mongabay.com/malaysia/images/borneo_6510.html) (дата обращения: 20.04.2013).



## Приложение А. Результаты первой схемы подбора параметров со многими диагональными элементами

Изображение	w <sub>1</sub>	w <sub>2</sub>	w <sub>3</sub>	w <sub>4</sub>	w <sub>5</sub>	q <sub>1</sub>	q <sub>2</sub>	q <sub>3</sub>	q <sub>4</sub>	q <sub>5</sub>	Ошибка
1	0.03947	0.23684	0.25000	0.23684	0.23684	0.90	0.89	0.91	0.90	0.89	1056
2	0.17000	0.17000	0.17000	0.32000	0.17000	0.89	0.80	0.90	0.90	0.91	1100
3	0.17408	0.15105	0.27079	0.23000	0.17408	0.70	0.88	0.91	0.89	0.91	1074
4	0.21000	0.08208	0.23597	0.23597	0.23597	0.89	0.73	0.93	0.90	0.81	933
5	0.23000	0.08000	0.23000	0.23000	0.23000	0.91	0.91	0.91	0.90	0.91	837
6	0.09081	0.21897	0.26125	0.21000	0.21897	0.85	0.82	0.98	0.91	0.90	678
7	0.20705	0.05000	0.29293	0.24296	0.20705	0.79	0.34	0.93	0.91	0.69	884
8	0.20483	0.00000	0.30675	0.26842	0.22000	0.83	0.33	0.87	0.88	0.55	732
9	0.19000	0.16590	0.16590	0.31229	0.16590	0.86	0.85	0.86	0.87	0.87	681
10	0.14108	0.32000	0.15600	0.24184	0.14108	0.33	0.42	0.84	0.86	0.86	578
11	0.18250	0.18250	0.18250	0.27000	0.18250	0.85	0.85	0.84	0.86	0.73	724
12	0.20805	0.12000	0.21746	0.24645	0.20805	0.89	0.91	0.91	0.91	0.91	561
13	0.16000	0.16000	0.36000	0.16000	0.16000	0.90	0.90	0.90	0.90	0.90	616
14	0.26000	0.01960	0.24013	0.24013	0.24013	0.64	0.98	0.95	0.83	0.81	686
15	0.23908	0.13856	0.21000	0.20618	0.20618	0.90	0.90	0.89	0.90	0.90	634
16	0.03025	0.19034	0.39264	0.19000	0.19677	0.88	0.93	0.89	0.89	0.79	420
17	0.12167	0.19595	0.25071	0.31000	0.12167	0.90	0.68	0.89	0.94	0.94	528
18	0.21701	0.21248	0.23307	0.25744	0.08000	0.92	0.86	0.90	0.83	0.88	602
19	0.20722	0.06239	0.23000	0.29317	0.20722	0.87	0.87	0.89	0.90	0.89	464
20	0.22502	0.21000	0.18833	0.18833	0.18833	0.90	0.88	0.89	0.90	0.86	663
21	0.16779	0.17407	0.31000	0.17407	0.17407	0.91	0.91	0.92	0.91	0.90	607
22	0.23750	0.05000	0.23750	0.23750	0.23750	0.59	0.91	0.91	0.91	0.91	648
23	0.07919	0.19991	0.21000	0.31098	0.19991	0.94	0.36	0.87	0.89	0.78	409
24	0.17657	0.24705	0.20000	0.18819	0.18819	0.92	0.67	0.97	0.91	0.89	630
25	0.00979	0.26000	0.24558	0.24232	0.24232	0.78	0.34	0.83	0.88	0.78	648
26	0.20512	0.09117	0.28859	0.21000	0.20512	0.86	0.38	0.87	0.89	0.85	758
27	0.01126	0.27868	0.27000	0.27868	0.16139	0.87	0.42	0.88	0.88	0.88	685
28	0.12370	0.20695	0.20695	0.22240	0.24000	0.82	0.82	0.91	0.86	0.85	454
29	0.03765	0.52000	0.14745	0.14745	0.14745	0.89	0.39	0.94	0.83	0.92	660
30	0.32143	0.25000	0.14286	0.14286	0.14286	0.34	0.66	0.90	0.89	0.90	781
31	0.14989	0.53524	0.14989	0.14000	0.02498	0.84	0.35	0.85	0.90	0.94	464
32	0.46501	0.01545	0.18528	0.17426	0.16000	0.49	0.84	0.87	0.90	0.88	724
33	0.15890	0.15890	0.37329	0.15890	0.15000	0.89	0.72	0.87	0.96	0.87	612
34	0.13398	0.13398	0.38807	0.21000	0.13398	0.85	0.90	0.86	0.89	0.89	587
35	0.02000	0.22022	0.22022	0.31935	0.22022	0.91	0.75	0.87	0.94	0.85	474
36	0.15300	0.24000	0.19173	0.26228	0.15300	0.87	0.63	0.92	0.88	0.86	599
37	0.12275	0.12275	0.47175	0.16000	0.12275	0.89	0.96	0.87	0.88	0.85	596
38	0.18070	0.15000	0.15389	0.36152	0.15389	0.80	0.90	0.91	0.90	0.89	662
39	0.01000	0.24750	0.24750	0.24750	0.24750	0.66	0.91	0.91	0.93	0.94	603
40	0.13988	0.14090	0.14719	0.41000	0.16204	0.89	0.90	0.90	0.90	0.90	766
41	0.15225	0.17000	0.37325	0.15225	0.15225	0.76	0.90	0.92	0.91	0.91	552
42	0.17077	0.12914	0.18932	0.34000	0.17077	0.88	0.90	0.90	0.88	0.71	719
43	0.15100	0.15100	0.31000	0.22025	0.16774	0.80	0.89	0.85	0.88	0.88	754
44	0.18000	0.18000	0.28000	0.18000	0.18000	0.78	0.67	0.88	0.94	0.50	561
45	0.18583	0.19806	0.19806	0.22000	0.19806	0.92	0.92	0.81	0.92	0.92	406
46	0.22000	0.03472	0.19720	0.35087	0.19720	0.93	0.88	0.94	0.95	0.62	648
47	0.16055	0.47000	0.16055	0.16055	0.04834	0.85	0.42	0.85	0.94	0.97	653
48	0.21692	0.21692	0.21692	0.32924	0.02000	0.91	0.55	0.95	0.89	0.75	474
49	0.19981	0.15000	0.19981	0.19981	0.25056	0.93	0.89	0.95	0.93	0.90	588

Изображение	w <sub>1</sub>	w <sub>2</sub>	w <sub>3</sub>	w <sub>4</sub>	w <sub>5</sub>	q <sub>1</sub>	q <sub>2</sub>	q <sub>3</sub>	q <sub>4</sub>	q <sub>5</sub>	Ошибка
50	0.14163	0.14163	0.44512	0.14163	0.13000	0.64	0.65	0.83	0.95	0.91	806
51	0.15426	0.15426	0.26000	0.27722	0.15426	0.88	0.83	0.86	0.94	0.89	566
52	0.46511	0.13000	0.14111	0.14111	0.12267	0.37	0.61	0.90	0.92	0.82	555
53	0.14169	0.15000	0.28679	0.27118	0.15035	0.85	0.85	0.87	0.95	0.93	474
54	0.26000	0.21793	0.21793	0.21793	0.08621	0.76	0.90	0.92	0.91	0.91	659
55	0.18306	0.18306	0.27083	0.18000	0.18306	0.85	0.88	0.86	0.85	0.85	488
56	0.17587	0.11000	0.17587	0.36240	0.17587	0.87	0.86	0.87	0.86	0.63	678
57	0.02013	0.24000	0.24662	0.24662	0.24662	0.85	0.75	0.90	0.89	0.59	603
58	0.19126	0.19126	0.26880	0.14869	0.20000	0.87	0.88	0.88	0.98	0.85	567
59	0.11013	0.20191	0.20191	0.26000	0.22606	0.86	0.82	0.94	0.84	0.83	631
60	0.16750	0.16750	0.16750	0.33000	0.16750	0.77	0.86	0.91	0.90	0.90	651
61	0.23250	0.23250	0.23250	0.23250	0.07000	0.88	0.68	0.82	0.89	0.88	908
62	0.04579	0.13000	0.27474	0.27474	0.27474	0.88	0.76	0.92	0.93	0.88	631
63	0.07896	0.22701	0.24000	0.22701	0.22701	0.79	0.88	0.87	0.88	0.88	714
64	0.23361	0.04918	0.23361	0.25000	0.23361	0.87	0.94	0.85	0.92	0.91	772
65	0.22676	0.23973	0.22676	0.22676	0.08000	0.68	0.54	0.88	0.90	0.88	581
66	0.17000	0.10986	0.38898	0.16558	0.16558	0.89	0.83	0.93	0.95	0.91	664
67	0.22069	0.14448	0.20390	0.15000	0.28093	0.69	0.84	0.80	0.94	0.84	563
68	0.04370	0.13042	0.35000	0.35326	0.12261	0.82	0.41	0.81	0.92	0.98	755
69	0.41184	0.06311	0.18000	0.13544	0.20961	0.35	0.48	0.92	0.92	0.86	578
70	0.12056	0.12056	0.30832	0.33000	0.12056	0.85	0.56	0.88	0.93	0.88	949
71	0.18462	0.18462	0.18462	0.24615	0.20000	0.90	0.89	0.90	0.92	0.86	883
72	0.23690	0.02931	0.23690	0.23690	0.26000	0.60	0.71	0.90	0.90	0.90	922
73	0.15000	0.14125	0.14125	0.42624	0.14125	0.59	0.87	0.93	0.91	0.95	717
74	0.17161	0.20946	0.20946	0.20946	0.20000	0.81	0.82	0.95	0.90	0.79	634
75	0.16795	0.16795	0.31614	0.18000	0.16795	0.85	0.84	0.84	0.87	0.86	809
76	0.21317	0.15048	0.21317	0.21000	0.21317	0.87	0.78	0.87	0.87	0.83	629
77	0.12219	0.12219	0.20000	0.43343	0.12219	0.73	0.79	0.95	0.83	0.88	468
78	0.20753	0.19517	0.19826	0.19903	0.20000	0.86	0.79	0.94	0.87	0.33	655
79	0.20313	0.20000	0.20313	0.20313	0.19060	0.90	0.90	0.94	0.90	0.90	708
80	0.11644	0.17000	0.19961	0.39751	0.11644	0.89	0.83	0.96	0.82	0.87	573
81	0.21000	0.21581	0.23134	0.25747	0.08538	0.80	0.46	0.88	0.88	0.99	631
82	0.16000	0.20348	0.22957	0.20348	0.20348	0.84	0.74	0.84	0.97	0.91	808
83	0.15965	0.18000	0.17779	0.30478	0.17779	0.92	0.95	0.89	0.91	0.81	552
84	0.20750	0.17000	0.20750	0.20750	0.20750	0.90	0.90	0.90	0.90	0.90	665
85	0.08987	0.60000	0.08987	0.13041	0.08987	0.63	0.37	0.92	0.89	0.62	773
86	0.19000	0.19000	0.24000	0.19000	0.19000	0.87	0.87	0.87	0.87	0.87	825
87	0.18172	0.18172	0.19000	0.23830	0.20826	0.85	0.87	0.94	0.88	0.79	775
88	0.25000	0.00000	0.25000	0.25000	0.25000	0.40	0.33	0.90	0.90	0.90	526
89	0.18843	0.08000	0.18843	0.35470	0.18843	0.88	0.46	0.92	0.89	0.86	845
90	0.44013	0.30000	0.09361	0.09966	0.06660	0.55	0.33	0.92	0.94	0.93	684
91	0.16637	0.34000	0.16089	0.16637	0.16637	0.45	0.34	0.88	0.96	0.82	691
92	0.12917	0.25719	0.14000	0.34447	0.12917	0.87	0.69	0.94	0.89	0.82	688
93	0.54414	0.06818	0.06818	0.18951	0.13000	0.48	0.46	0.98	0.92	0.72	589
94	0.12373	0.12373	0.25000	0.37880	0.12373	0.89	0.85	0.89	0.81	0.88	807
95	0.21250	0.21250	0.21250	0.21250	0.15000	0.92	0.92	0.87	0.93	0.87	609
96	0.21498	0.08505	0.27000	0.21498	0.21498	0.90	0.96	0.91	0.93	0.93	686
97	0.10928	0.20003	0.20035	0.29000	0.20035	0.89	0.86	0.87	0.85	0.88	780
98	0.18000	0.18000	0.28000	0.18000	0.18000	0.89	0.88	0.86	0.88	0.88	738
99	0.18000	0.19548	0.23356	0.19548	0.19548	0.87	0.87	0.87	0.87	0.87	560
100	0.09000	0.22750	0.22750	0.22750	0.22750	0.91	0.89	0.90	0.90	0.90	669
<b>Средняя</b>	<b>0.17645</b>	<b>0.17824</b>	<b>0.23122</b>	<b>0.23941</b>	<b>0.17467</b>	<b>0.804</b>	<b>0.745</b>	<b>0.894</b>	<b>0.899</b>	<b>0.849</b>	<b>668.65</b>
<b>СКО</b>	<b>0.09137</b>	<b>0.09986</b>	<b>0.07112</b>	<b>0.07174</b>	<b>0.05323</b>	<b>0.14707</b>	<b>0.197</b>	<b>0.038</b>	<b>0.033</b>	<b>0.102</b>	<b>----</b>

## Приложение Б. Средний процент ошибок валидации

Выбранное изображение	Подбор весов	Один диагональный элемент	1-ая схема со многими элементами	2-ая схема со многими элементами
1	2.1302	1.8788	1.866	1.8632
2	2.0526	1.8339	1.8317	1.8396
3	2.1	1.8287	1.8146	1.9035
4	2.1163	1.821	1.8306	1.8194
5	2.1385	1.8859	1.8899	1.8899
6	2.008	1.8611	1.9756	2.085
7	2.085	1.8152	1.8157	1.8295
8	2.0479	1.8907	1.8286	1.9219
9	2.0321	1.8934	1.8956	1.951
10	2.0982	1.8956	1.857	1.8853
11	2.1032	1.8729	1.9219	1.8435
12	2.1301	1.8807	1.8454	1.8807
13	2.0842	1.9004	1.9004	1.8981
14	2.1157	1.9237	1.8531	1.9004
15	2.093	1.9204	1.9032	1.9216
16	2.0651	1.9445	1.9933	1.8347
17	2.0793	1.8003	1.8302	1.8545
18	2.1311	1.8623	1.9378	1.866
19	2.0631	1.8694	1.8328	1.8303
20	2.0893	1.9054	1.9201	1.9611
21	2.1016	1.8949	1.8883	1.7958
22	2.0857	1.846	1.8356	1.8485
23	2.1129	1.9057	1.8362	1.8717
24	2.0782	1.858	1.907	1.9024
25	2.0886	1.9248	1.8567	1.9071
26	2.091	1.8862	1.8401	1.8703
27	2.073	1.9226	1.8027	1.8122
28	2.1243	1.8911	1.8598	1.842
29	2.0855	1.8739	1.8805	1.9499
30	2.1075	1.927	1.833	1.8649
31	2.1246	1.9284	1.8113	1.8931
32	2.0867	1.8426	1.8132	1.8488
33	2.0589	1.8062	1.8391	1.842
34	2.1109	1.8037	1.8327	1.7941
35	2.1099	1.9032	1.8933	1.9333
36	2.074	1.8366	1.7972	1.8306
37	2.0249	2.0645	1.9667	1.9789
38	2.1016	1.9005	1.8845	1.8694
39	2.1174	1.9255	1.9763	1.966
40	2.0793	1.9235	1.919	1.9235
41	2.0915	1.9506	1.9406	1.9272
42	2.0345	1.8027	1.8483	1.8553
43	2.0454	1.8398	1.8344	1.8391
44	2.1135	1.8548	1.8245	1.8022
45	2.1337	1.9275	1.9954	2.0845
46	2.0858	2.0532	2.0276	2.0483
47	2.0971	1.8682	1.835	1.8724
48	2.1046	1.8233	1.911	1.8485
49	2.081	1.8727	1.9344	1.8735
50	2.1447	1.8698	1.8449	1.8642
51	2.0294	1.8158	1.8245	1.8696

Выбранное изображение	Подбор весов	Один диагональный элемент	1-ая схема со многими элементами	2-ая схема со многими элементами
52	2.1139	1.8632	1.7834	1.9399
53	2.2094	1.8451	1.8326	1.8137
54	2.0747	1.8996	1.8529	1.8827
55	2.0542	1.9115	1.9052	1.8914
56	2.0879	1.9206	1.895	1.8934
57	2.0384	1.8228	1.8729	1.8556
58	2.0814	1.8748	1.8778	1.905
59	2.0462	1.9592	1.8297	1.9334
60	2.0479	1.8739	1.8416	1.8123
61	2.0367	1.9028	1.9188	1.7993
62	2.0819	1.846	1.8259	1.8495
63	2.0422	1.8473	1.8886	1.8987
64	2.0939	1.8879	1.8771	1.7957
65	2.0731	1.8934	1.8013	1.8238
66	2.0695	1.8343	1.8893	1.9226
67	2.1121	1.9065	1.9197	1.9065
68	2.1211	1.8827	1.8684	1.8595
69	2.0554	1.8501	1.8578	1.8625
70	2.1474	1.8013	1.7969	1.8262
71	2.0732	1.8638	1.8584	1.8327
72	2.0795	1.8501	1.8629	1.8728
73	2.0119	1.8328	1.9638	1.862
74	2.0816	1.8476	1.826	1.8428
75	2.0671	1.8486	1.8717	1.8995
76	2.0834	1.9304	1.8901	1.8626
77	2.116	1.8178	1.8285	1.7662
78	2.0944	1.8788	1.8724	1.8031
79	2.1372	1.8776	1.8841	1.9261
80	2.1627	1.8714	1.8276	1.8672
81	2.0663	1.8606	1.7919	1.8374
82	2.0569	1.921	1.8797	1.8807
83	2.1003	1.9394	1.9644	1.9473
84	2.0403	1.8792	1.8792	1.8578
85	2.1129	1.857	1.7892	1.8402
86	2.121	1.874	1.874	1.874
87	2.0986	1.9144	1.8222	1.9012
88	2.0502	1.9054	1.8633	1.8694
89	2.094	1.8984	1.8274	1.8319
90	2.0595	1.854	1.8347	1.892
91	2.0985	1.948	1.8753	2.0132
92	2.0406	1.9047	1.8398	1.8157
93	2.0728	1.9977	1.9687	1.8828
94	2.1193	1.8249	1.8171	1.8806
95	2.1452	1.9388	1.9503	1.9436
96	2.1124	1.9003	1.8745	1.869
97	2.0488	1.8937	1.8641	1.862
98	2.0966	1.8695	1.8766	1.8703
99	2.119	1.8879	1.8879	1.8879
100	2.0717	1.8317	1.8674	1.875
<b>Средняя (%)</b>	<b>2.0878</b>	<b>1.8817</b>	<b>1.8701</b>	<b>1.8785</b>