



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ М. В. ЛОМОНОСОВА

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

КАФЕДРА МАТЕМАТИЧЕСКИХ МЕТОДОВ ПРОГНОЗИРОВАНИЯ

ЮДИН НИКИТА ЕВГЕНЬЕВИЧ

Непрямое управление динамическим объектом

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:

к.ф-м.н., доцент

Гуров Сергей Исаевич

Москва, 2018

Содержание

1	Введение	3
2	Обзор предметной области	3
2.1	Обзор используемого оборудования	5
2.1.1	Microsoft Kinect	5
2.1.2	XhereraII	6
2.2	Постановка задачи	7
3	Описание экспериментов	8
3.1	Кросс-валидация	9
3.2	Использованные методы машинного обучения	10
3.2.1	SVM	10
3.2.2	Градиентный бустинг	11
3.2.3	Extra Trees	12
3.2.4	QDA	12
3.2.5	LDA	12
3.3	Сегментация кисти руки	13
3.3.1	SVM	18
3.3.2	QDA	18
3.3.3	Extra Trees	19
3.3.4	Градиентный бустинг	19
3.4	Сбор выборки	20
3.5	Признаковое представление сегментированной кисти руки	24
3.5.1	Гистограмма направлений градиентов (HOG)	24
3.5.2	Дефекты выпуклой оболочки	25
4	Вычислительные эксперименты	26
4.1	SVM	27
4.2	QDA	27
4.3	Extra Trees	27
4.4	Градиентный бустинг	28

4.5	Результаты экспериментов на тестовой выборке	28
4.6	Свёрточная нейронная сеть	29
4.7	Управление роботом	32
5	Заключение	32
	Список литературы	34
6	Приложение	38
6.1	Настройка гиперпараметров в задаче сегментации	38
6.1.1	SVM	38
6.1.2	QDA	39
6.1.3	Extra Trees	40
6.1.4	Градиентный бустинг	42
6.2	Настройка гиперпараметров в задаче классификации	43
6.2.1	SVM	43
6.2.2	QDA	45
6.2.3	Extra Trees	45
6.2.4	Градиентный бустинг	46

Введение

В современном мире нововведения в сфере вычислений и вычислительной техники требуют уже наличия более естественного интерфейса в интерактивных компьютерных системах, предназначенных для использования человеком. Человеко–компьютерное взаимодействие (HCI, human–computer interaction) — полидисциплинарное научное направление, существующее и развивающееся в целях совершенствования методов разработки, оценки и внедрения интерактивных компьютерных систем, предназначенных для использования человеком, а также в целях исследования различных аспектов этого использования [1].

Основной задачей данной работы является создание программного комплекса для управления механическим объектом (управляемый робот) при помощи визуальных команд, считываемых датчиком в составе интерфейса человеко–компьютерного взаимодействия, в данной работе в качестве визуальных команд рассмотрены жесты кисти правой руки, датчик регистрирует глубинное изображение. *Глубинным называется изображение, в котором значение каждого пикселя представляет собой в условных единицах расстояние от камеры до соответствующего пикселя объекта. В данной работе датчик, регистрирующий глубинное изображение, также называется глубинной камерой.*

Распознавание визуальных команд основано на классификации при помощи свёрточных нейронных сетей. Распознавание команд, выполняемых кистью правой руки, состоит из двух подзадач: сегментация глубинного изображения для определения области правой руки, классификация вида жеста для определения вида посылаемого сигнала роботу. Полученные в работе результаты приведены в разделах 3.3, 4.5, 4.6, 4.7. В качестве результатов отмечено видео с примером использования разработанного программного комплекса. Максимальная полученная точность классификации — 99%.

По результатам проделанной работы был сделан доклад на международной научной конференции студентов, аспирантов и молодых учёных «Ломоносов-2018».

Обзор предметной области

Одни из самых первых работ, посвящённых компьютерному зрению, появились в начале 50–х годов прошлого века. Началом эры компьютерного зрения можно считать работу Фрэнка Розенблатта, посвящённую модели перцептрона, опубликованную в 1957 г. [2]. Мо-

дель, построенная на принципах работы головного мозга, была настроена в рамках задачи обучения по прецедентам с учителем для распознавания простых геометрических фигур, таких как треугольник и квадрат. В работе Розенблатта фигурируют основные принципы моделирования с помощью искусственных нейронных сетей. Ещё одним ключевым моментом в становлении компьютерного зрения стали изыскания по анализу сцены в трёхмерном пространстве по двумерной проекции [3, 4]. Дальнейшее развитие в компьютерном зрении привело к анализу и разработке методов решения задачи семантической сегментации уже на реалистичных изображениях [5]. С начала 90-х годов развитие математического и вычислительного аппарата позволяет решать более сложные задачи анализа изображений, такие как распознавание лиц, происходит резкий скачок интереса и развитие статистического анализа в компьютерном зрении [6], трёхмерном моделировании [7]. За недавние десятилетия объёмы накопленных данных, используемых в задачах цифровой обработки изображений [8–11], а также достижения в области глубокого обучения позволили решать ещё более сложные задачи цифровой обработки и анализа изображений и видео [12, 13]. Также в последнее время в рамках парадигмы интернета вещей всё больше датчиков, позволяющих расширить возможности взаимодействия человека с окружающей средой, становится коммерчески доступным [14]. Ярким примером данного явления можно назвать появление системы Microsoft Kinect для взаимодействия с компьютером с помощью жестов [15]. Развитие непосредственно человеко–компьютерного взаимодействия происходило и происходит в тесном симбиозе с развитием самой вычислительной техники. Данное полидисциплинарное научное направление развивается в контексте разнонаправленных научных векторов: компьютерная графика, инженерная психология, эргономика, теория организации, когнитивистика, информатика и многие другие [16, 17]. За последние полвека исследования в области интерфейсов у Сазерленда, Ликлайдера, Энгельбарта, Кэй и Голдберг послужили появлением на свет многих привычных решений для взаимодействия человека с электронной вычислительной машиной: «мышь», поэлементно–адресуемое отображение (bitmap), концепция «окна», графическая оболочка, point–and–click–редакторы [18–21]. В последнее время всё большую актуальность приобретают интерфейсы, улучшающие эргономику взаимодействия в развивающихся технологиях, восстанавливающие утраченные возможности у человека [22, 23]. Краткая инфографика развития исследований некоторых решений и технологий в области интерфейсов представлена на рис. 1.

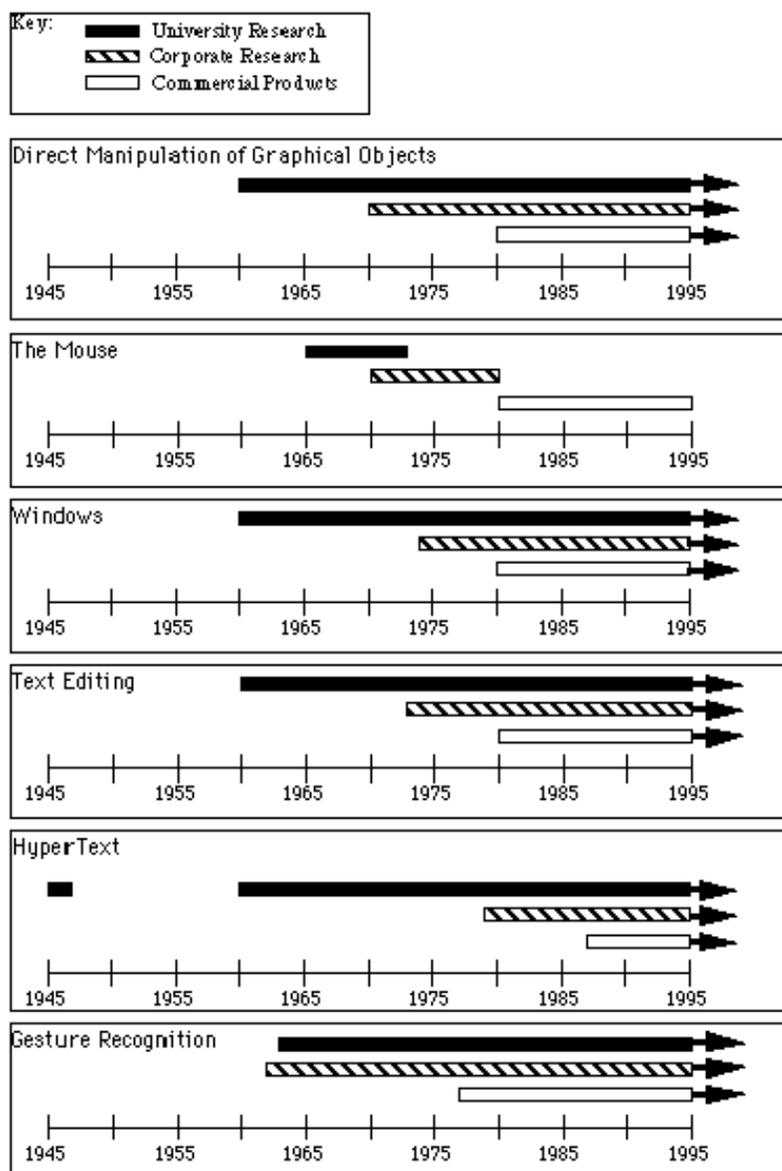


Рис. 1. Проведённые линии показывают время работы над указанными технологиями на разных уровнях: Direct Manipulation of graphical objects — управление с помощью графического пера, The Mouse — компьютерная «мышь», Windows — оконные интерфейсы, Text Editing — текстовые редакторы, HyperText — связь между документами ссылками внутри документа, Gesture Recognition — распознавание жестов с помощью различных датчиков[24]

Обзор используемого оборудования

Microsoft Kinect

Microsoft Kinect (рис. 2) — коммерческий датчик, основанный на добавлении периферийного устройства к игровой приставке Xbox 360 или к персональному компьютеру [15]. Разработан компанией Microsoft. Kinect позволяет пользователю взаимодействовать с основным устройством (Xbox 360, персональный компьютер) через устные команды, позы тела

и показываемые объекты или рисунки. Физически, Kinect — это горизонтально расположенная коробочка на небольшом круглом основании, которую помещают выше или ниже экрана. Размеры — примерно 23 см в длину и 4 см в высоту. Состоит из двух сенсоров глубины, цветной видеокамеры и микрофонной решетки. Микрофонная решетка позволяет производить локализацию источника звука и подавление шумов, что дает возможность говорить без наушников и микрофона. Датчик глубины состоит из инфракрасного проектора, объединенного с монохромной КМОП-матрицей (комплементарная структура металл-окись-полупроводник), что позволяет датчику Kinect получать трёхмерное изображение при любом естественном освещении. В Kinect датчик глубины способен корректно измерять расстояние от камеры до объекта в пределах от 0,4 м до 4 м.



Рис. 2. Внешний вид датчика Microsoft Kinect

KheperaII

KheperaII (рис. 3) — миниатюрный мобильный робот с функциональностью, схожей с более крупными роботами, которых используют в исследовании и в образовательном процессе [25]. Khepera изначально был разработан как исследовательский и образовательный инструмент швейцарской исследовательской приоритетной программы в Федеральной политехнической школе Лозанны. Робот позволяет в реальных условиях тестировать алгоритмы, разработанные в симуляции для планирования траектории, избегания препятствий, преобразования сенсорной информации, проверки гипотез о поведении, в частности. Данный робот представляет из себя комплекс, который может быть снабжён дополнительными модулями: видеокамерой, манипулятором для перемещения объектов, радиоантенной, и другими модулями. В своём распоряжении робот имеет 8 инфракрасных датчиков для ориентации в пространстве, расположенных практически равномерно по боковой поверхности робота. Робот

имеет форму «шайбы». Робот передвигается на двух колёсах, расположенных на диаметре «шайбы», плоскость колеса касается боковой поверхности робота.



Рис. 3. Внешний вид робота KheperaII

Постановка задачи

Рассматривается задача классификации. Дано пространство объектов произвольной природы \mathbb{X} и конечное множество ответов (меток) \mathbb{Y} , существует истинная целевая функция $y^* : \mathbb{X} \rightarrow \mathbb{Y}$, значение которой известно на конечной выборке объектов:

$$X^N = \{(x_1, y^*(x_1)), \dots, (x_N, y^*(x_N))\}, \quad x_i \in \mathbb{X}, \quad i = \overline{1, n}.$$

Требуется построить функцию (отображение, алгоритм) $y : \mathbb{X} \rightarrow \mathbb{Y}$, предсказывающую ответы для любых объектов из \mathbb{X} . Следующие определения крайне важны в данной работе:

- Признак объекта $x \in \mathbb{X}$ является $f : \mathbb{X} \rightarrow \mathbb{D}_f$, \mathbb{D}_f — множество значений признака f .
- Признаковое описание объекта $x \in \mathbb{X}$ в признаках f_i , $i = \overline{1, d}$ является:
 $\mathbf{x} = (f_1(x), \dots, f_d(x)) \in \mathbb{D} = \mathbb{D}_{f_1} \times \dots \times \mathbb{D}_{f_d}$, \mathbb{D} — признаковое пространство.

Функционал качества алгоритма y на выборке X^N для целевой функции y^* при известных истинных ответах на X^N и ответах алгоритма на X^N под названием общая точность

(accuracy) выглядит следующим образом:

$$accuracy(y, X^N) = \frac{1}{N} \sum_{i=1}^N [y^*(x_i) = y(x_i)].$$

[условие] — предикат, равный нулю при невыполнении условия, равный единице при выполнении условия. Так как рассмотренные в работе алгоритмы машинного обучения настраивают свои параметры, максимизируя значение функционала качества на обучающей выборке, то для выполнения оценки качества алгоритма y производится вычисление значения качества обученного алгоритма y на отложенной выборке, которая не была использована при обучении алгоритма.

Описание экспериментов

Первая часть решаемой задачи состоит в сегментации кисти правой руки на глубинном изображении. Данная подзадача в работе решена сведением задачи сегментации с двумя метками к задаче классификации каждого пикселя на два класса: $\mathbb{Y} = \{-1, 1\}$, где -1 соответствует пикселю фона изображения, 1 соответствует пикселю кисти руки изображения.

В данной работе модель пикселя глубинного изображения описана статистически (то есть рассмотрена задача обучения по прецедентам, в которой объект — пиксель, метка — целевая переменная). Поэтому задача классификации была решена в постановке из раздела 2.2. Выборка, использованная при решении задачи сегментации кисти руки была взята из репозитория по ссылке [26]. Выборка состоит из 5034 16-битных размеченных попиксельно глубинных изображений размера 640×480 , полученных с помощью датчика глубины Microsoft Kinect. Разметка состоит из двух классов: *кисть руки*, *фон*, доля пикселей класса *кисть руки* составляет 0.0286. Пример изображения из выборки представлен на рис. 4:

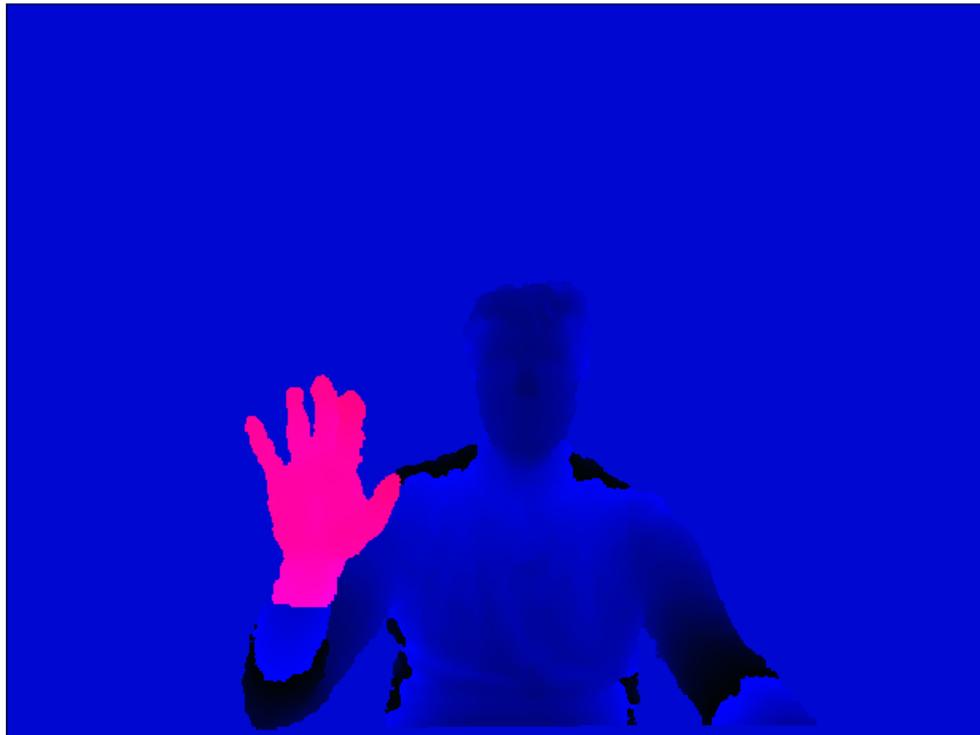


Рис. 4. 24-битное изображение из выборки. 16 бит хранят значение глубины в миллиметрах, 8 бит хранят номер класса

Кросс-валидация

В данной работе настройка гиперпараметров модели и оценка качества модели по имеющейся обучающей выборке происходили с помощью перекрёстной проверки на k подвыборках (k -fold Cross-validation). В данном методе оценки модели обучающая выборка разбивается на k непересекающихся практически одинаковых по размеру частей. В работе для каждого класса из обучающей выборки частота класса в каждой части из k была примерно равна частоте данного класса на всей обучающей выборке, то есть соблюдался баланс классов. В цикле по k подвыборкам каждая из k подвыборок ровно один раз используется для тестирования модели с измерением качества, на остальных $k - 1$ подвыборках модель обучается. В результате получается k значений качества, с помощью которых вычисляется выборочное среднее значения качества, являющееся оценкой качества модели, которую можно использовать для настройки гиперпараметров модели, выбирая среди доступных наборов гиперпараметров набор с соответствующим максимальным значением оценки качества модели.

Следующее определение является необходимым в данной работе:

Определение 1. *Сэмплирование — метод выбора подмножества наблюдаемых величин из данного множества, с целью выделения неких свойств исходного множества (отбор подвыборки из генеральной совокупности для оценки неких свойств генеральной совокупности).*

Использованные методы машинного обучения

В данной работе метод и соответствующий ему алгоритм, а также порождаемая методом модель имеют одинаковые названия.

SVM

Пусть:

$$X = \{(x_i, y_i)\}_{i=1}^N, \quad x_i \in \mathbb{R}^D, \quad y_i \in \{-1, +1\}, \quad C > 0, \quad \mathbf{a} = (a_1, \dots, a_N) \in \mathbb{R}^N,$$

$$k(x, y) = \exp\{-\gamma \cdot \|x - y\|_2^2\}, \quad \gamma > 0 \text{ — RBF ядро};$$

$$k(x, y) = \langle x, y \rangle \text{ — линейное ядро.}$$

В данной работе метод опорных векторов (Support Vector Machine, SVM [27]) решает поставленную задачу классификации, решая следующую оптимизационную задачу:

$$\begin{cases} \max_{\mathbf{a}} \sum_{n=1}^N a_n - \frac{1}{2} \cdot \sum_{n,m=1}^N a_n \cdot a_m \cdot y_n \cdot y_m \cdot k(x_n, x_m) \\ 0 \leq a_n \leq C, n = \overline{1, N} \\ \sum_{n=1}^N a_n \cdot y_n = 0 \end{cases}$$

Классификатор:

$$\hat{y}(x) = \text{sign} \left(\sum_{n=1}^N a_n \cdot y_n \cdot k(x_n, x) + b \right), \quad b = \frac{1}{N_{\bar{S}V}} \cdot \left(\sum_{j \in \bar{S}V} y_j - \sum_{j \in \bar{S}V} \sum_{i \in SV} a_i \cdot y_i \cdot k(x_i, x_j) \right),$$

sign — функция взятия знака числа, SV — множество индексов опорных объектов,

$\bar{S}V$ — множество индексов граничных опорных объектов (с зазором, равным 1), $N_{\bar{S}V}$ — мощность множества $\bar{S}V$. В многоклассовой задаче классификации ($y_i \in \{1, \dots, K\}$) метод опорных векторов применён сведением к серии из K задач бинарной классификации по схеме «один против всех»: $y_{ik} = +1$ — объект i принадлежит классу k , $y_{ik} = -1$ — объект i не при-

надлежит классу k . Подробнее о методе опорных векторов и его многоклассовом расширении можно прочесть в [28].

Градиентный бустинг

Пусть $X = \{(x_i, y_i)\}_{i=1}^N$, $x_i \in \mathbb{R}^D$, $y_i \in \{-1, +1\}$. В данной работе алгоритм бустинга решает поставленную задачу классификации следующим образом [29].

Пусть $f_j(x) \in \mathfrak{H}$ — решающее дерево [30]. Идея алгоритма состоит в построении линейной комбинации из $f_j(x)$ в ходе решения задачи регрессии. Пусть T — число листьев в дереве $f_t(x)$, $q(x)$ — функция, возвращающая индекс листа дерева, строящегося на итерации t , в который попал объект x , $I_j = \{i | i = \overline{1, N}, q(x_i) = j\}$, $w_j \in \mathbb{R}$ — ответ $f_t(x)$ в листе j , $f_t(x) = w_{q(x)}$, $\hat{y}_i^{t-1} = \sum_{k=1}^{t-1} f_k(x_i)$, $l(y_i, \hat{y}_i)$ — дважды дифференцируемая функция потерь, $\Omega(f_t)$ — регуляризатор. Тогда суммарные потери на выборке X на итерации t : $\mathfrak{L}^t = \sum_{i=1}^N l(y_i, \hat{y}_i^{t-1} + f_t(x_i)) + \Omega(f_t)$.

$$g_i = \frac{\partial l(y_i, \hat{y}_i^{t-1})}{\partial \hat{y}_i^{t-1}}, h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{t-1})}{\partial (\hat{y}_i^{t-1})^2}$$

$$\mathfrak{L}^t \simeq \sum_{i=1}^N (l(y_i, \hat{y}_i^{t-1}) + g_i \cdot f_t(x_i) + \frac{1}{2} \cdot h_i \cdot f_t^2(x_i)) + \Omega(f_t)$$

$$\bar{\mathfrak{L}}^t = \sum_{i=1}^N (g_i \cdot f_t(x_i) + \frac{1}{2} \cdot h_i \cdot f_t^2(x_i)) + \Omega(f_t)$$

$$\bar{\mathfrak{L}}^t = \sum_{i=1}^N (g_i \cdot f_t(x_i) + \frac{1}{2} \cdot h_i \cdot f_t^2(x_i)) + \gamma \cdot T + \frac{\lambda}{2} \sum_{j=1}^T w_j^2 =$$

$$\sum_{j=1}^T \left(\sum_{i \in I_j} g_i \cdot w_j + \frac{1}{2} \sum_{i \in I_j} (h_i + \lambda) \cdot w_j^2 \right) + \gamma \cdot T.$$

В дереве f_t оптимальные веса имеют вид:

$$w_j = - \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}, \quad j = \overline{1, T}.$$

Комитет строится итеративно, на каждой итерации строится дерево жадным образом: каждый узел строится с точки зрения минимизации $\bar{\mathfrak{L}}^t$, если текущий узел имел бы только листья в качестве потомков. Алгоритм построения комитета останавливается по заданному критерию останова.

Классификатор — $\hat{y}(x) = \text{sign} \left(\sum_{m=1}^M f_m(x) \right)$. В многоклассовой задаче классификации ($y_i \in \{1, \dots, K\}$) строится K описанных выше комитетов, каждый из кото-

рых оценивает вероятность принадлежности i -го объекта к классу k :

$$p_k(x) = \frac{1}{1 + \exp\left(-\sum_{m=1}^M f_{km}(x)\right)},$$

используются принципы схемы «один против всех»: $y_{ik} = +1$ — объект i принадлежит классу k , $y_{ik} = -1$ — объект i не принадлежит классу k , соответственно для объекта x модель в качестве ответа выдаёт наиболее вероятный класс относительно $\{p_k(x)\}_{k=1}^K$.

Extra Trees

В данной работе Extra Trees использован для решения задачи классификации[31]. Классификатор представляет собой множество отдельных решающих деревьев. Каждое из деревьев строится по выборке того же размера, что и исходная выборка, полученной из исходной путем случайного выбора с возвращением. Ответом данного комитета является мода ответов входящих в комитет решающих деревьев. В комитете Extra Trees при построении дерева каждый узел дерева определяется по случайно выбранному подмножеству признаков, вместо поиска наиболее оптимального значения порога на подмножестве признаков для обучающей выборки наиболее оптимальное значение порога выбирается из множества сэмплированных порогов для данного узла относительно подмножества признаков и данной обучающей выборки. Подробнее о решающих деревьях можно прочитать в [30].

QDA

Пусть $X = \{(x_i, y_i)\}_{i=1}^N$, $x_i \in \mathbb{R}^D$, $y_i \in \{1, \dots, K\}$, $w \in \mathbb{R}^D$, λ_y — цена ошибки на объекте класса y , P_y — априорная вероятность класса y , μ_y — математическое ожидание вектора класса y , $\hat{\Sigma}_y = (1 - \alpha) \cdot \Sigma_y + \alpha \cdot I$ — регуляризованная ковариационная матрица класса y , α — параметр регуляризации ковариационной матрицы класса y : Σ_y . В данной работе метод классификации квадратичный дискриминант (Quadratic Discriminant Analysis, QDA) решает поставленную задачу классификации следующим образом[32]:

$$\hat{y}(x) = \arg \max_y \left\{ \ln\{\lambda_y \cdot P_y\} - \frac{1}{2} \cdot (x - \mu_y)^T \cdot \hat{\Sigma}_y^{-1} \cdot (x - \mu_y) - \frac{1}{2} \cdot \ln \left| \hat{\Sigma}_y \right| \right\}.$$

LDA

Пусть $X = \{(x_i, y_i)\}_{i=1}^N$, $x_i \in \mathbb{R}^D$, $y_i \in \{1, \dots, K\}$, $w \in \mathbb{R}^D$, λ_y — цена ошибки на объекте класса y , P_y — априорная вероятность класса y , μ_y — математическое ожидание вектора класса y , Σ_y — ковариационная матрица класса y . В данной работе метод классификации

линейный дискриминант (Linear Discriminant Analysis, LDA) решает поставленную задачу классификации следующим образом[32]:

$$\hat{y}(x) = \arg \max_y \{ \ln\{\lambda_y \cdot P_y\} - \frac{1}{2} \cdot \mu_y^T \cdot \Sigma_y^{-1} \cdot \mu_y + x^T \cdot \Sigma_y^{-1} \cdot \mu_y \}.$$

Сегментация кисти руки

В решении задачи сегментации кисти руки на глубинном изображении для каждого пикселя x (соответствующий ему вектор координат на изображении также обозначен за x) предложено следующее признаковое описание, являющееся модификацией признакового описания в [33], признаковое описание обозначено в рамке с формулой (1).

$$f_{i(\mathbf{u})}(I, x) = d_I \left(\left\lfloor x + \frac{\mathbf{u}}{d_I(x)} \right\rfloor \right) - d_I(x) \quad (1)$$

- x — вектор координат пикселя
- \mathbf{u} — вектор смещения
- I — изображение
- $d_I(x)$ — глубина пикселя x на изображении I
- $i(\mathbf{u})$ — порядковый номер признака \mathbf{u}

Главный параметр данного признакового описания: множество векторов \mathbf{u} . Оно было определено из следующей эвристики: координаты векторов \mathbf{u} нужно так выбирать, чтобы была возможность оценить локальные геометрические особенности и геометрические особенности всей сцены для каждого пикселя. Поэтому значения каждой координаты были выбраны так, чтобы в логарифмическом масштабе они могли образовать арифметическую прогрессию для каждой координаты отдельно, что и отражено в таблицах 1 и 2:

Полжительные и отрицательные смещения по горизонтали, $\pm u_i$															
± 1	± 2	± 3	± 4	± 5	± 6	± 8	± 10	± 14	± 19	± 25	± 34	± 45	± 60	± 81	± 108

Таблица 1. Значения координаты вектора смещения для столбца

Полжительные и отрицательные смещения по вертикали, $\pm u_j$											
± 1	± 2	± 3	± 4	± 5	± 8	± 11	± 15	± 21	± 29	± 40	± 57

Таблица 2. Значения координаты вектора смещения для строки

Во избежание выхода за границы исходного изображения при вычислении признаков проводилось зеркальное отражение изображения вдоль границ. Множество векторов смещения представлено множеством $\{\pm u_i\}_{i=1}^{16} \times \{\pm u_j\}_{j=1}^{12}$, то есть всего 768 признаков.

Так как мощность выборки большая (1546444800 объектов), то чтобы обучить модель за разумное время, в работе была организована схема перевыборки для отбора необходимой подвыборки. Оценив экспертно вычислительные мощности, размер необходимой подвыборки был выбран примерно равным 5000000. В качестве возможных схем перевыборки были предложены следующие эвристические стратегии:

1. Провести суперпикселизацию изображения: выполнить кластеризацию изображения, разбив множество пикселей изображения с предложенным признаковым описанием (формула (1)) на 1000 кластеров, минимизируя суммарное квадратичное отклонение точек кластеров от центров своих кластеров (в данной задаче достаточно найти локальный минимум). Из каждого кластера случайно выбрать пиксель, соблюдая баланс классов (равенство долей пикселей класса *кисть руки* в исходной выборке и в отобранной подвыборке) относительно текущего изображения: среди кластеров, в которых пиксели были из разных классов внутри кластера, происходило перераспределение для сохранения баланса классов: для каждого такого кластера выбирался случайно пиксель только из одного класса, класс выбирался случайно, но баланс классов соблюдался.
2. Равномерное сэмплирование пикселей внутри каждого класса, баланс классов соблюдался.
3. Неравномерное сэмплирование пикселей внутри каждого класса, баланс классов соблюдался. В неравномерном сэмплировании пиксель $x|c$ попадает в выборку с вероятностью $p(x|c) = \frac{256 - d_I(x)}{Z_c}$, где Z_c — нормировочная константа, c — номер класса, соотношение классов соблюдалось.

В соответствии с описанием стратегии **1** кластеризация пикселей одного изображения была выполнена с помощью алгоритма K-means [34]. В результате проведённого эксперимента

было выяснено, что стратегия **1** работает крайне медленно на матрице выборки пикселей одного изображения (рис. 5) «объекты–признаки» размера 307200×768 , в которой по строкам перечислены объекты, по столбцам — признаки. Поэтому данная стратегия не была использована. Эмпирически было выявлено, что стратегия **1** чаще сэмплирует точки из областей границ объектов (рис. 6).

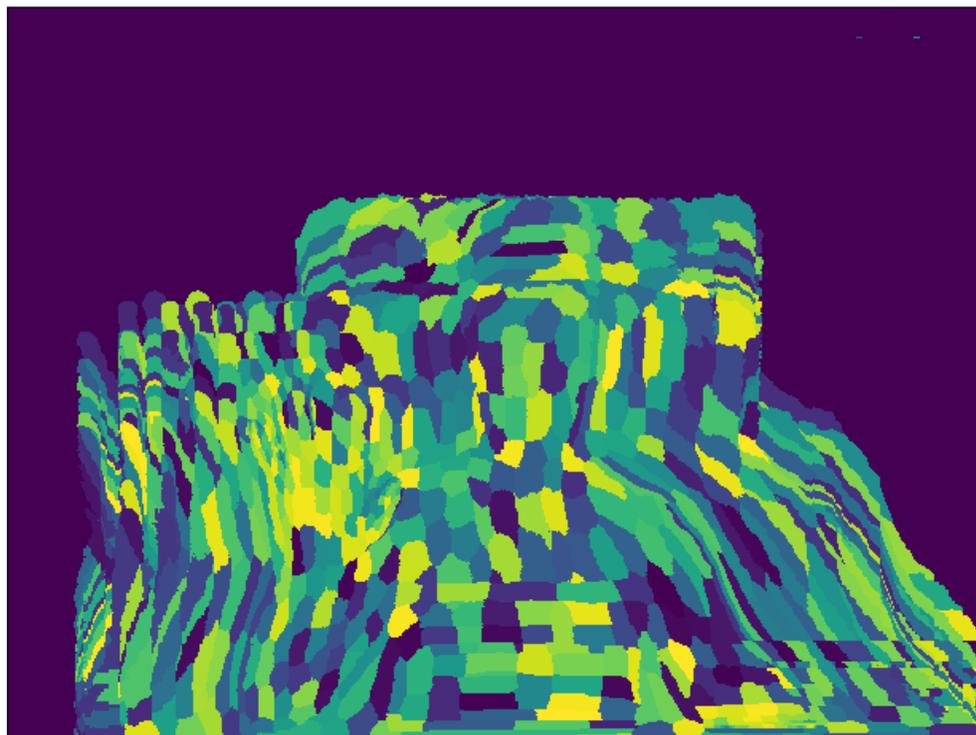


Рис. 5. Результат суперпикселизации одного изображения. Каждый цвет отвечает за свой кластер.

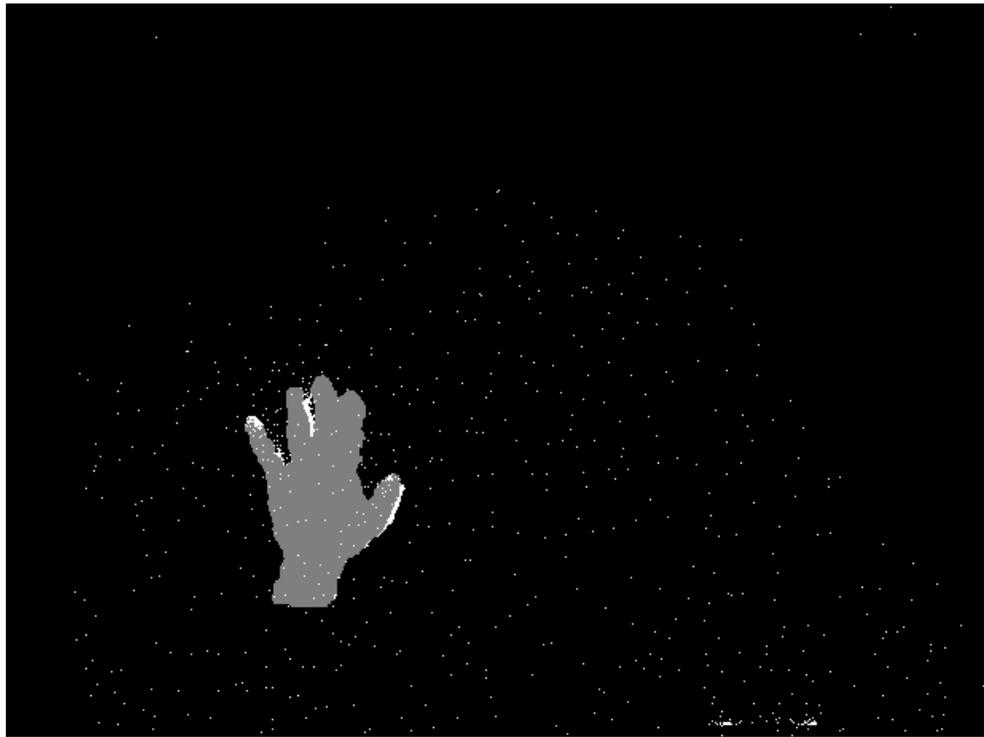


Рис. 6. Результат сэмплирования из изображения после суперпикселизации. Белый цвет — выбранный пиксель. Серый цвет — маска первого класса (пиксели первого класса)



Рис. 7. Результат равномерного сэмплирования из изображения. Белый цвет — выбранный пиксель. Серый цвет — маска первого класса (пиксели первого класса)

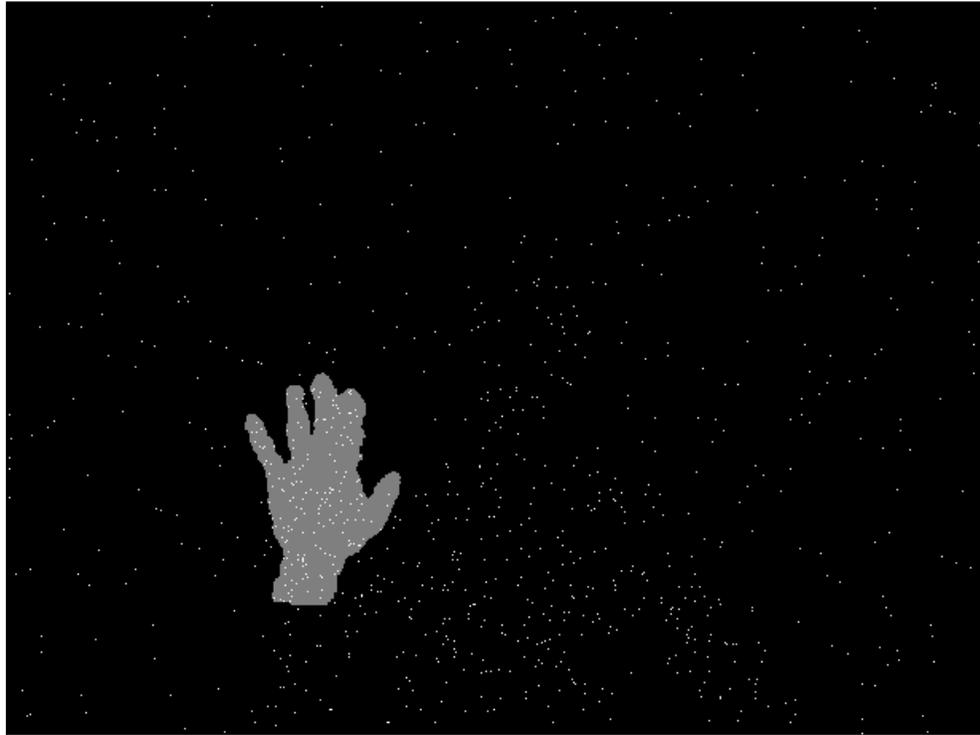


Рис. 8. Результат неравномерного сэмплирования из изображения. Белый цвет — выбранный пиксель. Серый цвет — маска первого класса (пиксели первого класса)

Примеры сэмплирования с помощью стратегии **2** и стратегии **3** представлены на рис. 7 и 8 соответственно. В исходной выборке из 5034 глубинных изображений были выделены 100 изображений для тестовой выборки, оставшаяся выборка изображений рассматривается как обучающая выборка, на которой были настроены с помощью кросс-валидации гиперпараметры рассмотренных алгоритмов машинного обучения. На всей выборке было единое признаковое представление пикселей (формула (1)). В качестве алгоритма, решающего задачу кластеризации через классификацию был испытан ряд алгоритмов машинного обучения: SVM, градиентный бустинг, Extra Trees, QDA, LDA. Гиперпараметры данных алгоритмов были настроены с помощью процедуры кросс-валидации при числе подвыборок, равном 8. Значения гиперпараметров перебирались на сетках.

С помощью стратегий **2** и **3** были собраны независимо по одной подвыборке для каждой стратегии из оставшейся обучающей выборки после разделения исходной выборки на обучающую и тестовую выборки. Соответственно, собранная с помощью стратегии **2** подвыборка не пересекается с тестовой выборкой, аналогично собранная с помощью стратегии **3** подвыборка не пересекается с тестовой выборкой.

Критерий качества при настройке гиперпараметров алгоритмов в задаче сегментации был выбран, исходя из следующей стратегии: в результате сегментации изображения большинство пикселей, назначенных как пиксель кисти руки, должны являться пикселями кисти руки в действительности. Соответственно, за критерий качества была выбрана *точность*:

$$precision(y, X^N) = \frac{\sum_{i=1}^N [1 = y(x_i)] \cdot [1 = y^*(x_i)]}{\sum_{i=1}^N [1 = y(x_i)]}.$$

Для каждой стратегии (**2** и **3** стратегии) и её соответствующей обучающей выборки была проведена процедура настройки гиперпараметров рассмотренных алгоритмов с помощью процедуры кросс-валидации. Ниже представлены детали проведённой процедуры кросс-валидации.

SVM

Графики, отображающие зависимость качества на кросс-валидации представлены в приложении: рис. 23, 24.

	равномерное сэмплирование		неравномерное сэмплирование	
	C	γ	C	γ
Линейный SVM	1,0	—	1,0	—
RBF SVM	0,1	0,0001	1,0	0,001

Таблица 3. Оптимальные гиперпараметры

QDA

Графики, отображающие зависимость качества на кросс-валидации представлены в приложении: рис. 25, 26.

	равномерное сэмплирование	неравномерное сэмплирование
	α	α
QDA	1,0	1,0

Таблица 4. Оптимальные гиперпараметры

Extra Trees

Графики, отображающие зависимость качества на кросс-валидации представлены в приложении: рис. 27, 28, 29, 30.

	равномерное сэмплирование		неравномерное сэмплирование	
	максимальная глубина дерева	количество деревьев	максимальная глубина дерева	количество деревьев
Extra Trees	25	36	25	40

Таблица 5. Оптимальные гиперпараметры

Градиентный бустинг

Графики, отображающие зависимость качества на кросс-валидации представлены в приложении: рис. 31, 32.

	равномерное сэмплирование		неравномерное сэмплирование	
	максимальная глубина дерева	количество деревьев	максимальная глубина дерева	количество деревьев
Градиентный бустинг	9	120	9	120

Таблица 6. Оптимальные гиперпараметры

В результате рассмотренных выше экспериментов были настроены алгоритмы машинного обучения для использования стратегий **2** и **3**. При этом использование стратегии **3** (неравномерное сэмплирование) позволило на тестовой выборке из 100 изображений решить незначительно лучше задачу, чем использование стратегии **2** (равномерное сэмплирование), что отражено в таблице 7:

Точность решения задачи сегментации кисти руки на тестовой выборке		
	равномерное сэмплирование	неравномерное сэмплирование
QDA	0,55	0,58
LDA	0,61	0,72
Линейный SVM	0,53	0,60
RBF SVM	0,37	0,41
Градиентный бустинг	0,98	0,99
Extra trees	0,98	0,99

Таблица 7. Значения точности на тестовой выборке

Лучшее качество на тестовой выборке показали алгоритмы Extra Trees и градиентный бустинг. Финальная версия алгоритма Extra Trees, выполняющего сегментацию пикселей глубинного изображения, была обучена с помощью стратегии **3**, гиперпараметры данного алгоритма были настроены по кросс-валидации на выборке, собранной с помощью стратегии **3**.

Сбор выборки

С помощью финальной версии алгоритма сегментации Extra Trees был проведён сбор выборки сегментированных изображений кисти правой руки, размеченной на 5 классов, соответствующих командам робота: по 20000 изображений на каждый класс. Кисть руки была выделена алгоритмом заливки (псевдокод представлен на листинге 1), начиная с пикселя кисти руки с глубиной, равной медиане глубины пикселей кисти руки на изображении, данная эвристика предложена для увеличения робастности процесса сбора выборки, начальный пиксель алгоритмом сегментации обозначен как пиксель кисти руки. Пиксели кисти руки определены алгоритмом Extra Trees. При сборе выборки было выполнено следующее условие: для датчика Microsoft Kinect ближайший пиксель относится к телу человека.

Algorithm 1 Алгоритм заливки (x^0, a, b, I, M)

Вход — начальный пиксель x^0 , нижний порог $a \geq 0$, верхний порог $b \geq 0$, глубинное изображение I , изображение–маска кисти руки M (перед первым вызовом процедуры проинициализирована нулями, кроме одной начальной точки)

for $x \in$ множество соседей пикселя x^0 **do**
 if $(d_I(x^0) - a \leq d_I(x) \leq d_I(x^0) + b) \& (d_M(x) = 0)$ **then**
 $d_M(x) := 1$
 $M :=$ Алгоритм заливки (x, a, b, I, M)

Output изображение–маска кисти руки M

После работы алгоритма заливки строится ограничивающий множество пикселей кисти руки прямоугольник, который вместе с внутренней областью трактуется как глубинное изображение. Данное изображение билинейной интерполяцией приводится к изображению размера 80×80 , завершается предобработка изображения в собираемой выборке, иллюстрация этапов процесса сегментации представлена на рис. 12. Стоит заметить, что уже предобработанные изображения размера 80×80 линейным преобразованием с округлением переводятся из 16–битного представления в 8–битное.

Таким образом, после сбора выборки сегментированных кистей правых рук будет готова сбалансированная выборка, описывающая 5 классов:

0. *команда стоп* — данный класс соответствует бездействию робота KheperaII.
1. *команда назад* — данный класс соответствует движению назад робота KheperaII.
2. *команда налево* — данный класс соответствует движению налево робота KheperaII.
3. *команда направо* — данный класс соответствует движению направо робота KheperaII.
4. *команда вперёд* — данный класс соответствует движению вперёд робота KheperaII.

Примеры глубинных изображений из собранной выборки сегментированных кистей правых рук представлены на рис. 16, 19.



Рис. 9. входное изображение



Рис. 10. сегментированная с помощью Extra Trees и заливки кисть руки



Рис. 11. отмасштабированная билинейной интерполяцией кисть руки

Рис. 12. Иллюстрация процесса сегментации кисти руки (рис. 9, 10, 11)



Рис. 13. сегментированное изображение 0 класса

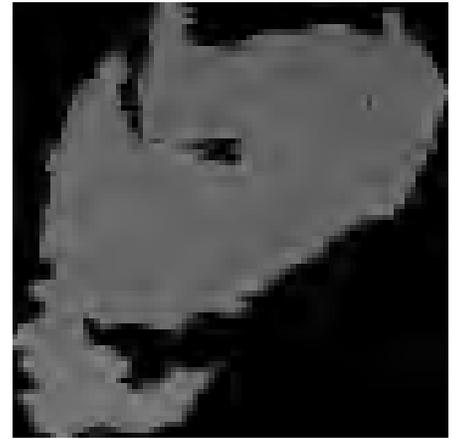


Рис. 14. сегментированное изображение 1 класса

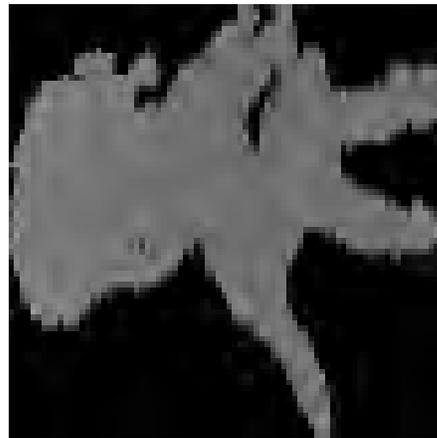


Рис. 15. сегментированное изображение 2 класса

Рис. 16. Примеры изображений собранной выборки кисти правой руки

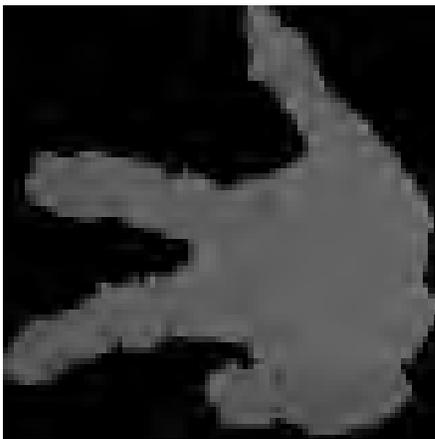


Рис. 17. сегментированное изображение 3 класса



Рис. 18. сегментированное изображение 4 класса

Рис. 19. Примеры изображений собранной выборки кисти правой руки

Признаковое представление сегментированной кисти руки

В качестве признакового описания глубинного изображения кисти руки рассмотрены предложенные в данной работе дефекты выпуклой оболочки и гистограммы направлений градиентов [35].

Гистограмма направлений градиентов (НОГ)

Для вычисления признакового описания у входящего изображения вычисляются градиенты для каждого пикселя с помощью свёртки с оператором Собеля[36]. Формула (2) определяет операторы Собеля для вычисления горизонтальной (S_x) и вертикальной (S_y) составляющей градиента изображения. В данной работе на следующем этапе происходит разбиение изображения на непересекающиеся блоки пикселей размером 8×8 , в каждом таком блоке происходит вычисление гистограммы направлений градиентов. Каждый пиксель в блоке участвует во взвешенном голосовании для каналов гистограммы направлений, основанном на значении градиентов. Каналы гистограммы равномерно распределяются от 0 до 360 градусов (в данной работе — 8 каналов). Далее блоки группируются без пересечений в более крупные связные блоки (4 соседних блока 8×8 в один блок 16×16), в которых происходит нормировка каналов гистограмм. Итоговое признаковое представление является вытянутым в один вектор набором каналов гистограмм. Пример визуализации признакового описания НОГ представлен на рис. 20.

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2)$$



Рис. 20. Признаковое описание сегментированной кисти руки с помощью НОГ (чем ярче полоса в блоке, тем больше сумма абсолютных длин градиентов, перпендикулярных полосе в блоке)

Дефекты выпуклой оболочки

Признаковое описание под названием дефекты выпуклой оболочки является гистограммой направлений векторов (в аналогии с НОГ) точек контура, наиболее удалённых от соответствующих отрезков выпуклой оболочки контура (наиболее удалённая точка контура от соответствующего отрезка выпуклой оболочки и соответствующий отрезок выпуклой оболочки называются таковыми, если относятся к одному «желобу» в контуре, как показано на рис. 21). Для отрезка выпуклой оболочки соответствующая наиболее удалённая точка контура принадлежит ломаной контура, имеющей концы в концах данного отрезка выпуклой оболочки, и никакой другой отрезок выпуклой оболочки не содержит в себе оба конца данной ломаной. Векторы откладываются от центра изображения. Но, в отличие от НОГ, в данном признаковом описании блок один, в нём 128 каналов, равномерно распределённых от 0 до 360 градусов, взвешенная сумма по каналам основывается на расстоянии от точки до соответствующего отрезка выпуклой оболочки: чем больше значение в канале, тем больше суммарное расстояние от точек до соответствующих им отрезков выпуклой оболочки вдоль соответствующего канала направления векторов соответствующих точек. Предложенное признаковое описание основано на морфологии пальцев руки и на центральном поло-

жении кисти руки почти на каждом изображении в собранной выборке сегментированной кисти правой руки.



Рис. 21. Светло-серый цвет — выпуклая оболочка, белый цвет — наиболее удалённые от выпуклой оболочки точки контура, серый цвет — объект

Контур объекта на изображении в оттенках серого найден алгоритмом, описанным в [37], выпуклая оболочка вычисляется итерацией по вершинам контура в соответствии с алгоритмом, описанным в [38].

Вычислительные эксперименты

Собранная выборка из глубинных изображений сегментированной кисти правой руки была разбита на стратифицированные выборки: обучающая выборка размером 80000 объектов, тестовая выборка размером 20000 объектов. В следующих экспериментах настройка гиперпараметров моделей машинного обучения происходила по схеме кросс-валидации с 8 подвыборками для каждого набора гиперпараметров из сетки, на которой происходит поиск оптимальных значений гиперпараметров, если не указано иное. В экспериментах были использованы следующие модели машинного обучения: SVM, градиентный бустинг, Extra Trees, QDA, LDA, свёрточная нейронная сеть (CNN) [39, 40]. В качестве критерия качества была рассмотрена общая точность.

SVM

Результаты экспериментов с моделью SVM представлены на рис. 33, 34, 35, в таблице 8. Значения критерия качества получены с помощью кросс-валидации.

	HOG признаки		дефекты выпуклой оболочки	
	C	γ	C	γ
линейный SVM	0,001	—	0,0001	—
SVM с RBF ядром	1	0,1	1000	0,001

Таблица 8. Значения оптимальных гиперпараметров для метода и признакового описания

QDA

Результаты экспериментов с моделью QDA представлены на рис. 36, в таблице 9. Значения критерия качества получены с помощью кросс-валидации.

	HOG признаки	дефекты выпуклой оболочки
	α	α
QDA	0,05	0

Таблица 9. Значения оптимальных гиперпараметров в QDA для признакового описания

Extra Trees

В данном эксперименте гиперпараметры оптимизировались независимо. Результаты экспериментов с моделью Extra Trees представлены на рис. 37, 38, в таблицах 10, 11. Значения критерия качества получены с помощью кросс-валидации.

Максимальная глубина дерева		
	HOG признаки	дефекты выпуклой оболочки
Extra Trees	24	25

Таблица 10. Значения оптимальной максимальной глубины дерева в Extra Trees для признакового описания

Количество деревьев		
	НОГ признаки	дефекты выпуклой оболочки
Extra Trees	100	20

Таблица 11. Значения оптимального количества деревьев в Extra Trees для признакового описания

Градиентный бустинг

Результаты экспериментов с моделью градиентного бустинга представлены на рис. 39, 40, в таблице 12. Значения критерия качества получены с помощью кросс-валидации.

	НОГ признаки		дефекты выпуклой оболочки	
	максимальная глубина дерева	количество деревьев	максимальная глубина дерева	количество деревьев
Градиентный бустинг	9	120	9	120

Таблица 12. Значения оптимальных гиперпараметров в градиентном бустинге для признакового описания

Результаты экспериментов на тестовой выборке

Как видно из результатов, представленных в таблице 13, рассмотренные настроенные методы с предложенными признаками (дефекты выпуклой оболочки) по общей точности показывают меньшее значение в решаемой задаче, чем точно такие же методы, но с признаковым описанием НОГ. При этом, лучше поставленную задачу решил алгоритм градиентного бустинга.

Общая точность на тестовой выборке		
	НОГ	дефекты выпуклой оболочки
Линейный SVM	0,54	0,39
RBF SVM	0,56	0,46
QDA	0,76	0,72
LDA	0,79	0,78
Градиентный бустинг	0,93	0,85
Extra Trees	0,82	0,72

Таблица 13. Значения качества настроенных методов на тестовой выборке для признакового описания

Свёрточная нейронная сеть

Нейросетевая модель в себя включает следующие слои:

- **Свёрточный слой (Conv)**. На вход принимает тензор

$$X = (x_1, \dots, x_{C_1}) \in \mathbb{R}^{C_1 \times H_1 \times W_1},$$

выполняет операции свёртки \star и сложения с тензорами

$$K = (k_{ij})_{i=1, j=1}^{C_2, C_1} \in \mathbb{R}^{C_2 \times C_1 \times h \times w}, \quad B = (b_1, \dots, b_{C_2}) \in \mathbb{R}^{C_2 \times H_2 \times W_2},$$

вычисляя тензор $Y = (y_1, \dots, y_{C_2}) \in \mathbb{R}^{C_2 \times H_2 \times W_2}$:

$$y_i = b_i + \sum_{j=1}^{C_1} x_j \star k_{ij}, \quad i = \overline{1, C_2};$$

в качестве функции активации рассмотрена

$$ReLu(x) = z; \quad x = (x_{ijk})_{i,j,k=1}^{C,H,W}, \quad z = (z_{ijk})_{i,j,k=1}^{C,H,W} \in \mathbb{R}^{C \times H \times W}, \quad z_{ijk} = \max\{0, x_{ijk}\}.$$

Параметры h, w обычно называют размерами ядра, за C_i обычно обозначают число «каналов». Свёрточный слой возвращает $ReLu(Y)$.

- **Слой субдискретизации (Max–pooling).** Производит нелинейное уплотнение входящего тензора $X = (x_1, \dots, x_{C_1}) \in \mathbb{R}^{C_1 \times H_1 \times W_1}$ для формирования $Y = (y_1, \dots, y_{C_1}) \in \mathbb{R}^{C_1 \times H_2 \times W_2}$. В данной работе для каждого x_i происходило разбиение на непересекающиеся блоки 2×2 , каждый из которых трактовался как элемент y_i , в качестве значения элемента выбирался максимальный в блоке элемент. Слой субдискретизации возвращает Y .
- **Полносвязный слой.** На вход принимает вектор $x \in \mathbb{R}^{D_1}$, выполняет операции матричного умножения с $W \in \mathbb{R}^{D_2 \times D_1}$ и сложения с вектором $b \in \mathbb{R}^{D_2}$, вычисляя вектор $y \in \mathbb{R}^{D_2}$:

$$y = W \cdot x + b;$$

в качестве функций активации рассмотрены $ReLU(x)$ и

$$Softmax(x) = z; \quad x = (x_m)_{m=1}^k, \quad z = (z_m)_{m=1}^k \in \mathbb{R}^k, \quad z_i = \frac{e^{x_i}}{\sum_{m=1}^k e^{x_m}}.$$

Полносвязный слой возвращает $ReLU(y)$ или $Softmax(y)$.

- **Стохастический слой Dropout[41].** Производит следующее преобразование вектора $x \in \mathbb{R}^k$: $y = x \circ \xi$, \circ — произведение Адамара; $y \in \mathbb{R}^k$, $\xi = (\xi_i)_{i=1}^k$, $\xi_i \stackrel{i.i.d.}{\sim} Bernoulli(1-p)$. В данной работе был определён гиперпараметр p . Величина $p = 0.4$ была подобрана на сетке с помощью кросс-валидации с 8 подвыборками на обучающей выборке. Стохастический слой Dropout возвращает y .

При построении нейросетевой модели, основанной на свёрточных слоях, была использована следующая эвристика: первые слои свёрточной нейронной сети состоят из слоёв, выполняющих операции свёртки с функцией активации $f(x) = \max\{0, x\}$, субдискретизации (max–pooling), которые будут выполнять функцию сжатия изображения, которое в сжатом виде уже будет подано на вход полносвязным слоям. На основе данной эвристики была предложена модель, схематически изображённая на рис. 22. Стохастический слой Dropout выполняет функцию регуляризатора. В качестве признакового описания глубинного изображения сегментированной кисти правой руки выступает само изображение.

Построенная нейронная сеть была обучена двумя способами. Первый способ подразумевает настройку весов нейронной сети на обучающей выборке из 80000 объектов. В рам-

ках второго способа обучающая выборка из 80000 изображений была расширена до 1120000 изображений через порождение новых изображений: порождённые изображения кисти руки отличаются от оригинального изображения кисти руки сдвигом глубины кисти руки на равномерно распределённую случайную величину, в ходе процесса порождения изображений происходил контроль нахождения каждого пикселя в области допустимых значений. Второй способ обучения связан с эмпирическими наблюдениями, подтверждающими улучшение обобщающей способности с увеличением размера обучающей выборки[42], и является проверкой данной гипотезы в рамках решаемой задачи.

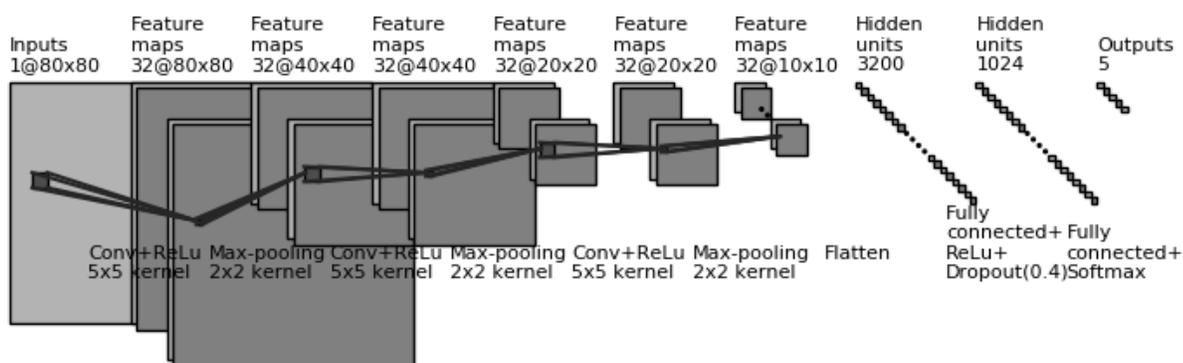


Рис. 22. Архитектура построенной нейронной сети

Результаты экспериментов со свёрточной нейронной сетью представлены в таблице 14:

Размер обучающей выборки	Общая точность на тестовой выборке
80000	0.997
1120000	0.999

Таблица 14. Значения качества свёрточной нейронной сети

Результаты эксперимента продемонстрировали практически отсутствие изменения качества на тестовой выборке вследствие расширения обучающей выборки, но данный факт не противоречит изложенным выше эмпирическим заключениям. По сравнению с результатами, изложенными в таблице 13, результаты в таблице 14 выше по качеству, что означает превосходство нейросетевой модели с исходными признаками (глубинное изображение) над другими рассмотренными в данной работе моделями в рамках решаемой задачи.

Управление роботом

Была написана программа на языке программирования C++ для считывания показаний с датчика Microsoft Kinect, распознавания показаний в реальном времени, выделения управляющих команд и их передачи роботу KheperaII. Программа способна передавать 5 видов команд, описанных в разделе 3.4. Распознавание команд достаточно точное (около 99% по общей точности, модель — свёрточная нейронная сеть). В качестве одной из мер, повышающих робастность предсказания, служит отправка команды не раньше, чем через 30 кадров после предыдущей отправки команды. При этом, посылается команда, наиболее чаще предсказываемая за прошедшие 30 кадров. С примером управления роботом записано видео[43].

Заключение

В данной работе было проведено исследование по распознаванию определённых жестов кисти руки на показаниях, полученных с помощью камеры глубины. В работе были предложены стратегии по организации перевыборочных планов для обучения модели классификатора, среди предложенных стратегий была выбрана наиболее оптимальная, с помощью которой были обучены модели, сегментирующие пиксели глубинного изображения, среди обученных моделей была выбрана оптимальная.

С помощью обученной оптимальной сегментирующей модели была собрана выборка глубинных изображений кисти правой руки. Для решения задачи распознавания жеста кисти руки были рассмотрены различные признаковые описания и алгоритмы машинного обучения, лучшей комбинацией по общей точности оказалась свёрточная нейронная сеть на исходных признаках, данная модель достигла общей точности на тестовой выборке, равной 99%.

Результатом работы стал программный комплекс, написанный на языке программирования C++, связавший систему Microsoft Kinect с роботом KheperaII. Предложенная система позволяет управлять с помощью жестов роботом, контролируя перемещение робота на плоскости. В данной задаче достигается высокая общая точность, что позволяет утверждать о решении поставленной задачи.

В ходе выполнения выпускной квалификационной работы:

- 1) Были исследованы способы организации перевыборочного плана.

- 2) Были исследованы методы обработки цифровых изображений.
- 3) Рассмотрены несколько алгоритмов машинного обучения применительно к задаче сегментации пикселей глубинного изображения и к задаче распознавания жестов на глубинном изображении.
- 4) Разработан программный комплекс по управлению роботом при помощи глубинной камеры.
- 5) По результатам работы сделан доклад на международной научной конференции студентов, аспирантов и молодых учёных «Ломоносов–2018» [44].

Список литературы

1. ACM SIGCHI curricula for human-computer interaction / Thomas T Hewett, Ronald Baecker, Stuart Card [и др.]. ACM, 1992.
2. Rosenblatt Frank. The perceptron, a perceiving and recognizing automaton Project Para. Cornell Aeronautical Laboratory, 1957.
3. Roberts Lawrence G. Machine perception of three-dimensional solids. Ph.D. thesis: Massachusetts Institute of Technology. 1963.
4. Guzmán Adolfo. Decomposition of a visual scene into three-dimensional bodies // Proceedings of the December 9-11, 1968, fall joint computer conference, part I / ACM. 1968. С. 291–304.
5. Ohta Yu-ichi, Kanade Takeo, Sakai Toshiyuki. An analysis system for scenes containing objects with substructures // Proceedings of the Fourth International Joint Conference on Pattern Recognitions. 1978. С. 752–754.
6. Turk Matthew A, Pentland Alex P. Face recognition using eigenfaces // Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on / IEEE. 1991. С. 586–591.
7. RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments / Peter Henry, Michael Krainin, Evan Herbst [и др.] // In the 12th International Symposium on Experimental Robotics (ISER / Citeseer. 2010.
8. Gradient-based learning applied to document recognition / Yann LeCun, Léon Bottou, Yoshua Bengio [и др.] // Proceedings of the IEEE. 1998. Т. 86, № 11. С. 2278–2324.
9. Krizhevsky Alex, Hinton Geoffrey. Learning multiple layers of features from tiny images. 2009.
10. Krizhevsky Alex, Sutskever Ilya, Hinton Geoffrey E. Imagenet classification with deep convolutional neural networks // Advances in neural information processing systems. 2012. С. 1097–1105.
11. Jain Suyog Dutt, Grauman Kristen. Active Image Segmentation Propagation. // CVPR. Т. 1. 2016. с. 4.

12. Deng Li, Yu Dong [и др.]. Deep learning: methods and applications // Foundations and Trends® in Signal Processing. 2014. Т. 7, № 3–4. С. 197–387.
13. LeCun Yann, Bengio Yoshua, Hinton Geoffrey. Deep learning // nature. 2015. Т. 521, № 7553. с. 436.
14. Internet of Things (IoT): A vision, architectural elements, and future directions / Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic [и др.] // Future generation computer systems. 2013. Т. 29, № 7. С. 1645–1660.
15. Zhang Zhengyou. Microsoft kinect sensor and its effect // IEEE multimedia. 2012. Т. 19, № 2. С. 4–10.
16. Теоретические и практические основы человеко-компьютерного взаимодействия: базовые понятия человеко-компьютерных систем в информатике и информационной безопасности / А П Фисун, Л А Гращенко, В В Митяев [и др.]. 2004.
17. Пископсель А А, Щедровицкий Л П. Инженерная психология и эргономика. 1996.
18. Sutherland Ivan. SKETCHPAD-a man-machine graphical interface. Ph.D. thesis: PhD thesis, MIT. 1963.
19. Licklider Joseph CR. Man-computer symbiosis // IRE transactions on human factors in electronics. 1960. № 1. С. 4–11.
20. Engelbart Douglas C. Augmenting human intellect: a conceptual framework (1962) // PACKER, Randall and JORDAN, Ken. Multimedia. From Wagner to Virtual Reality. New York: WW Norton & Company. 2001. С. 64–90.
21. Goldberg Adele. Educational uses of a dynabook // Computers & Education. 1979. Т. 3, № 4. С. 247–266.
22. Azuma Ronald T. A survey of augmented reality // Presence: Teleoperators & Virtual Environments. 1997. Т. 6, № 4. С. 355–385.
23. Wiederhold B, Riva G. Balance recovery through virtual stepping exercises using Kinect skeleton tracking: a followup study with chronic stroke patients // Annual Review of Cybertherapy and Telemedicine 2012: Advanced Technologies in the Behavioral, Social and Neurosciences. 2012. Т. 181. С. 108–112.

24. Myers Brad A. A brief history of human-computer interaction technology // interactions. 1998. T. 5, № 2. С. 44–54.
25. Michel Olivier. Cyberbotics Ltd. Webots™: professional mobile robot simulation // International Journal of Advanced Robotic Systems. 2004. T. 1, № 1. с. 5.
26. Real-time continuous pose recovery of human hands using convolutional networks / Jonathan Tompson, Murphy Stein, Yann Lecun [и др.] // ACM Transactions on Graphics (ToG). 2014. T. 33, № 5. с. 169.
27. Cortes Corinna, Vapnik Vladimir. Support-vector networks // Machine learning. 1995. T. 20, № 3. С. 273–297.
28. Weston Jason, Watkins Chris. Multi-class support vector machines: Tech. Rep.: : Citeseer, 1998.
29. Chen Tianqi, Guestrin Carlos. Xgboost: A scalable tree boosting system // Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining / ACM. 2016. С. 785–794.
30. Classification and regression trees / Leo Breiman, Jerome Friedman, Charles J Stone [и др.]. CRC press, 1984.
31. Geurts Pierre, Ernst Damien, Wehenkel Louis. Extremely randomized trees // Machine learning. 2006. T. 63, № 1. С. 3–42.
32. Friedman Jerome, Hastie Trevor, Tibshirani Robert. The elements of statistical learning. Springer series in statistics New York, 2001. T. 1.
33. Real-time human pose recognition in parts from single depth images / Jamie Shotton, Andrew Fitzgibbon, Mat Cook [и др.] // Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on / Ieee. 2011. С. 1297–1304.
34. Arthur David, Vassilvitskii Sergei. k-means++: The advantages of careful seeding // Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms / Society for Industrial and Applied Mathematics. 2007. С. 1027–1035.

35. Dalal Navneet, Triggs Bill. Histograms of oriented gradients for human detection // Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on / IEEE. Т. 1. 2005. С. 886–893.
36. Методы компьютерной обработки изображений / В А Сойфер, М В Гашников, Н И Глу-мов [и др.]. 2003.
37. Suzuki Satoshi [и др.]. Topological structural analysis of digitized binary images by border following // Computer vision, graphics, and image processing. 1985. Т. 30, № 1. С. 32–46.
38. Sklansky Jack. Finding the convex hull of a simple polygon // Pattern Recognition Letters. 1982. Т. 1, № 2. С. 79–83.
39. Backpropagation applied to handwritten zip code recognition / Yann LeCun, Bernhard Boser, John S Denker [и др.] // Neural computation. 1989. Т. 1, № 4. С. 541–551.
40. Deep learning / Ian Goodfellow, Yoshua Bengio, Aaron Courville [и др.]. MIT press Cambridge, 2016. Т. 1.
41. Dropout: A simple way to prevent neural networks from overfitting / Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky [и др.] // The Journal of Machine Learning Research. 2014. Т. 15, № 1. С. 1929–1958.
42. Wang Jason, Perez Luis. The effectiveness of data augmentation in image classification using deep learning: Tech. Rep.: : Technical report, 2017.
43. Видео с примером управления роботом. <https://drive.google.com/open?id=1B6Kd0eFY-FjVipL4AablQLkX3N5magot>. Дата актуализации: 2018-05-08.
44. Юдин Н. Непрямое управление динамическим объектом // Сборник тезисов XXV между-народной научной конференции студентов, аспирантов и молодых учёных «Ломоносов–2018». МАКС Пресс, 2018.

Приложение

Настройка гиперпараметров в задаче сегментации

SVM

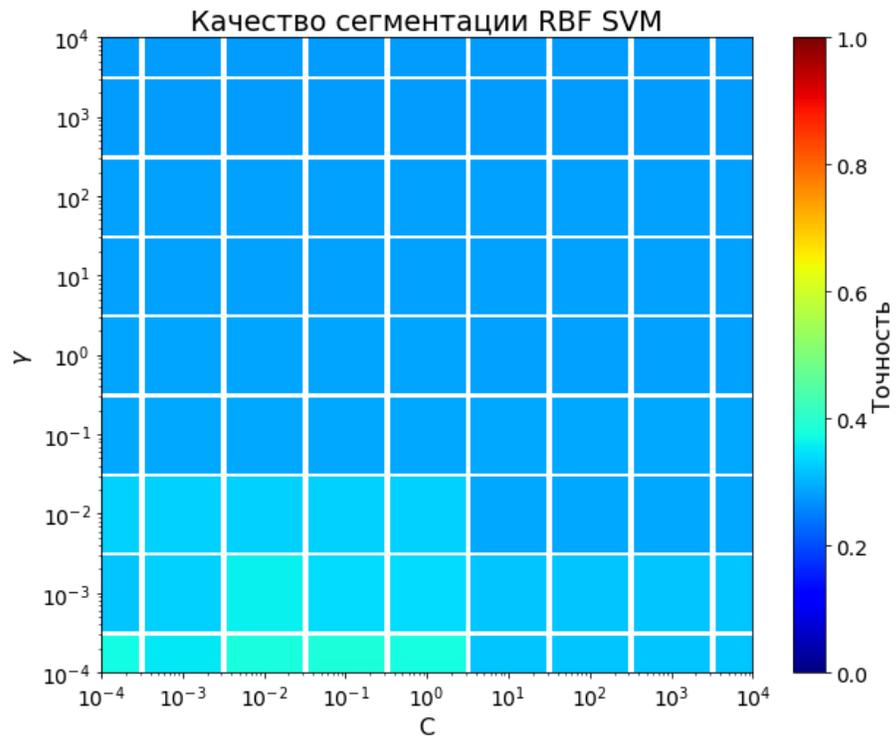


Рис. 23. Поиск оптимальных гиперпараметров метода SVM по критерию точности в случае равномерно сэмплированной выборки

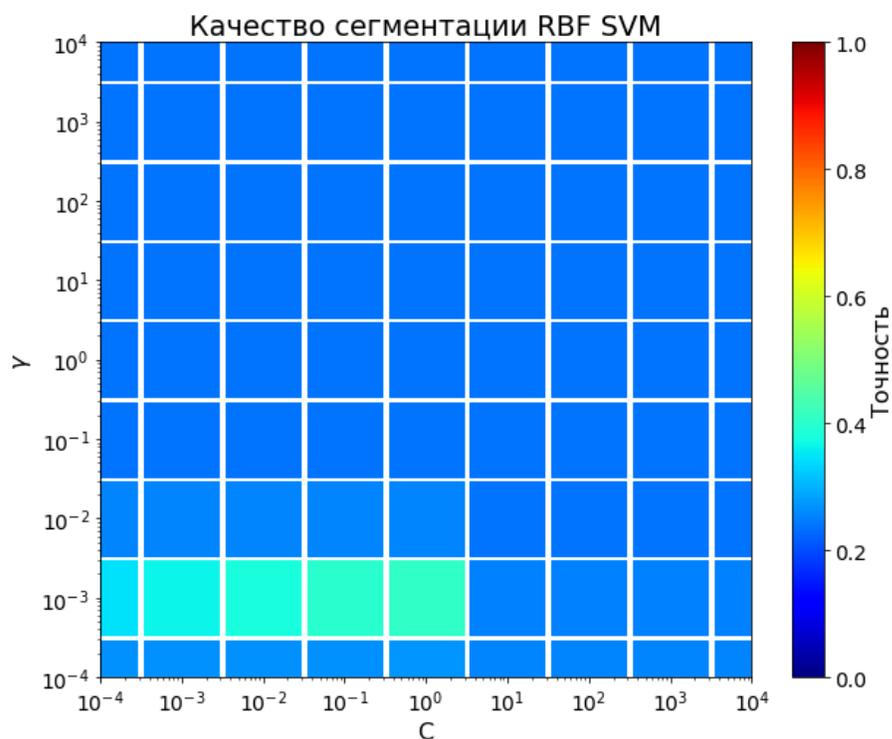


Рис. 24. Поиск оптимальных гиперпараметров метода SVM по критерию точности в случае неравномерно сэмплированной выборки

QDA

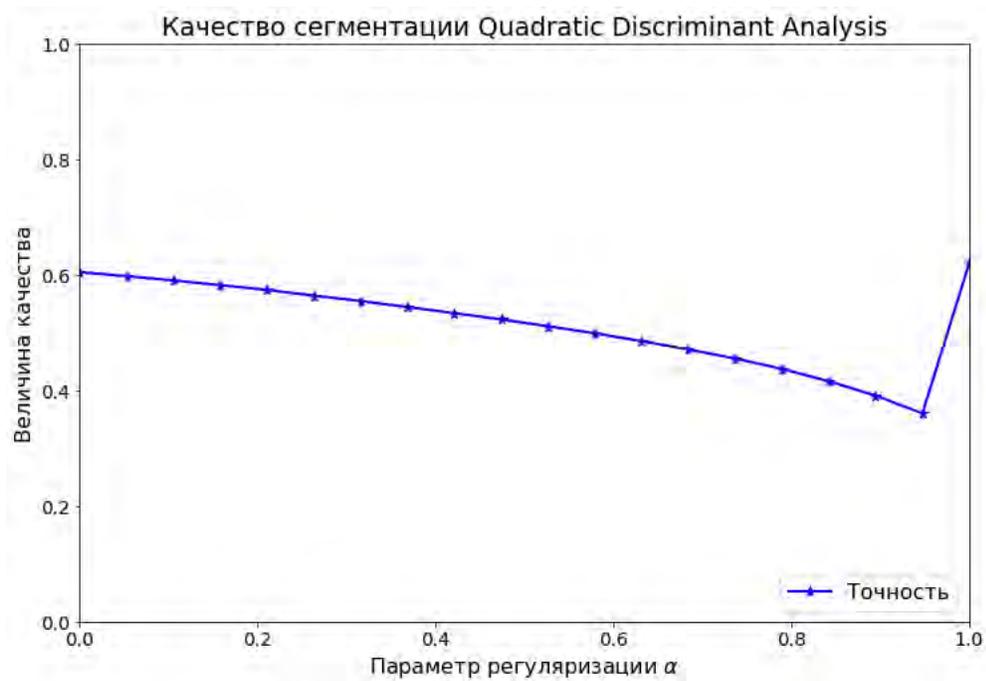


Рис. 25. Поиск оптимальных гиперпараметров метода QDA по критерию точности в случае равномерно сэмплированной выборки

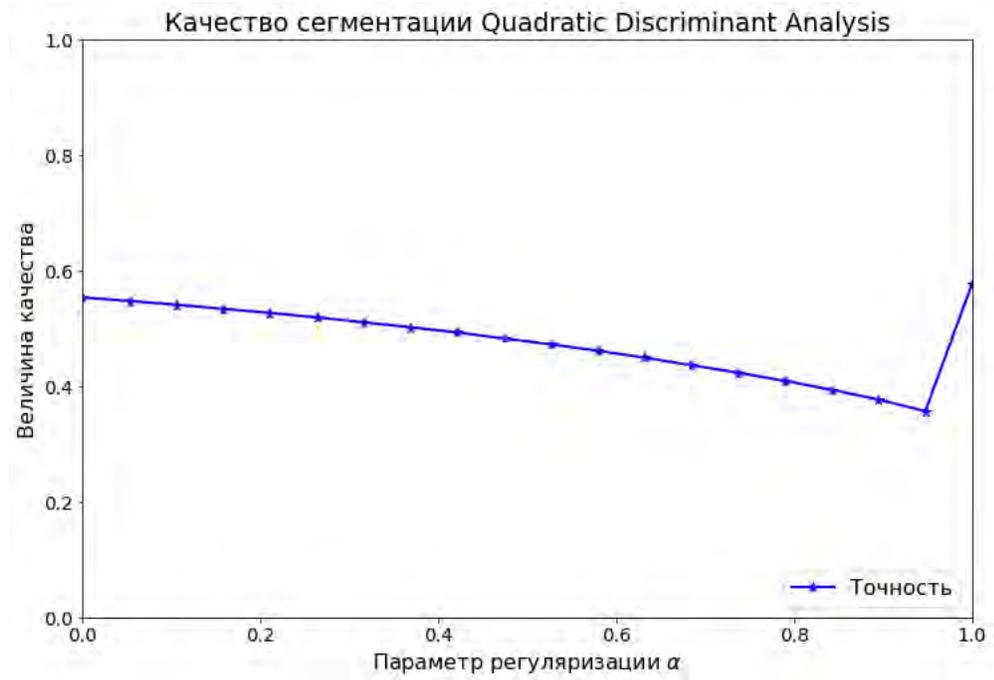


Рис. 26. Поиск оптимальных гиперпараметров метода QDA по критерию точности в случае неравномерно сэмплированной выборки

Extra Trees

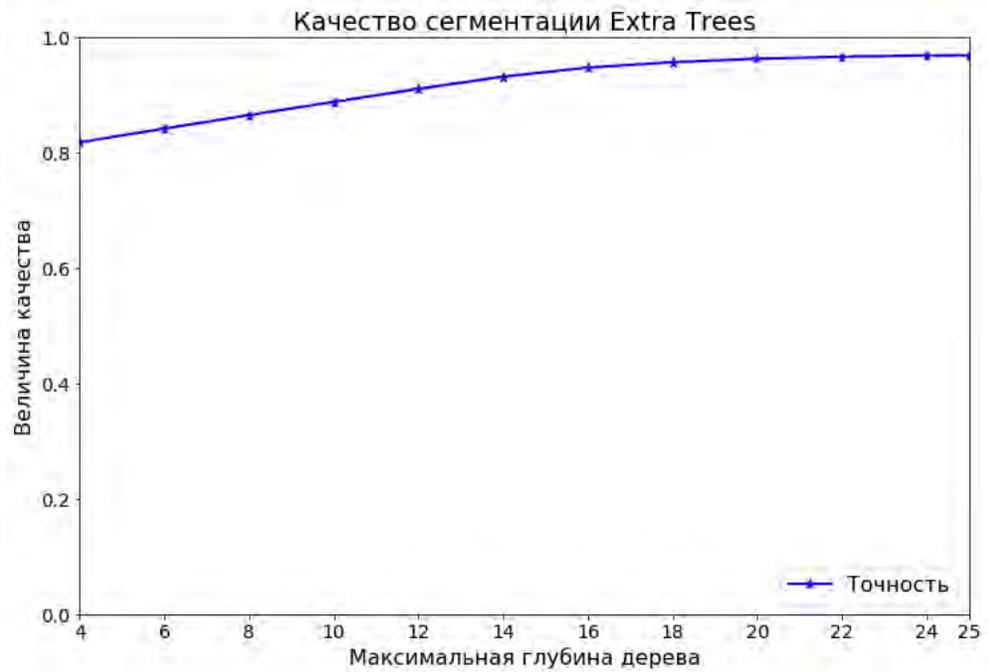


Рис. 27. Поиск оптимальной глубины дерева (число деревьев равно 32) в случае неравномерно сэмплированной выборки

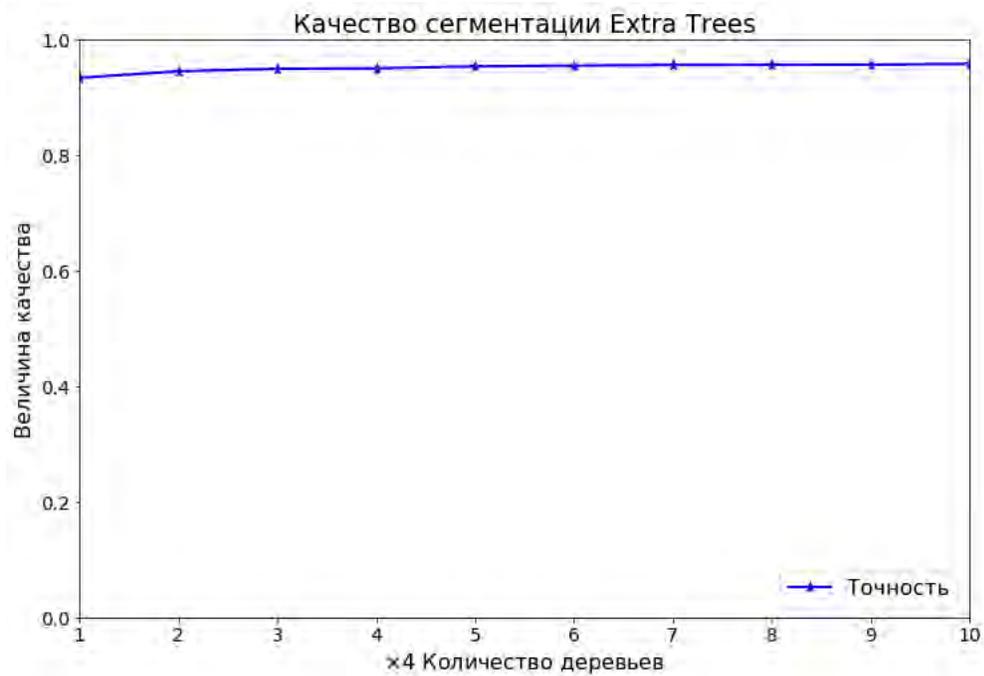


Рис. 28. Поиск оптимального числа деревьев (максимальная глубина дерева равна 18) в случае неравномерно сэмплированной выборки

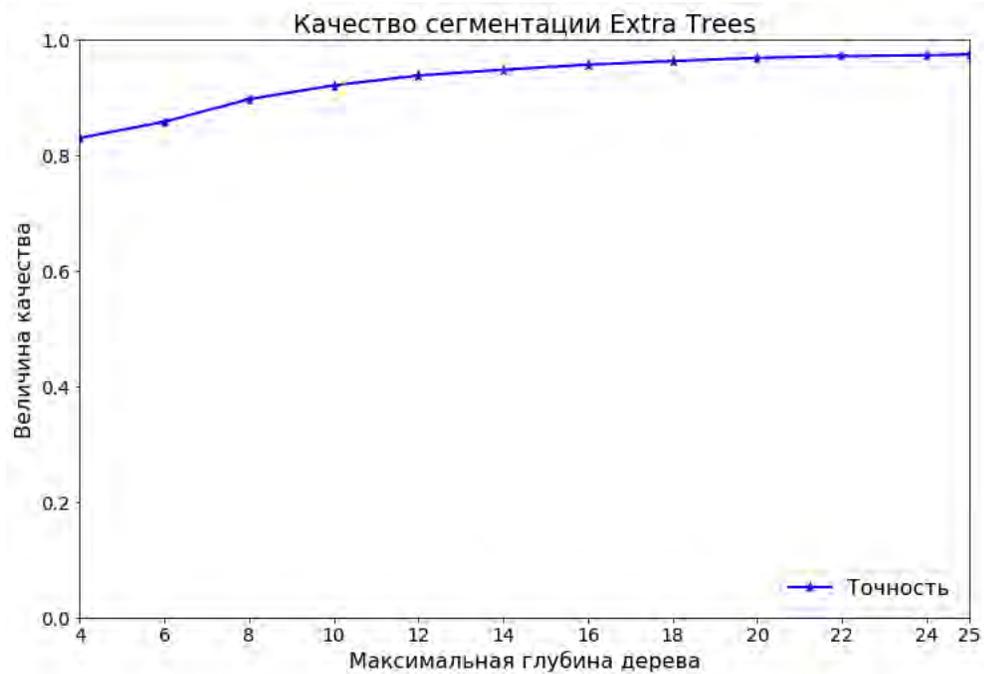


Рис. 29. Поиск оптимальной глубины дерева (количество деревьев равно 32) в случае равномерно сэмплированной выборки

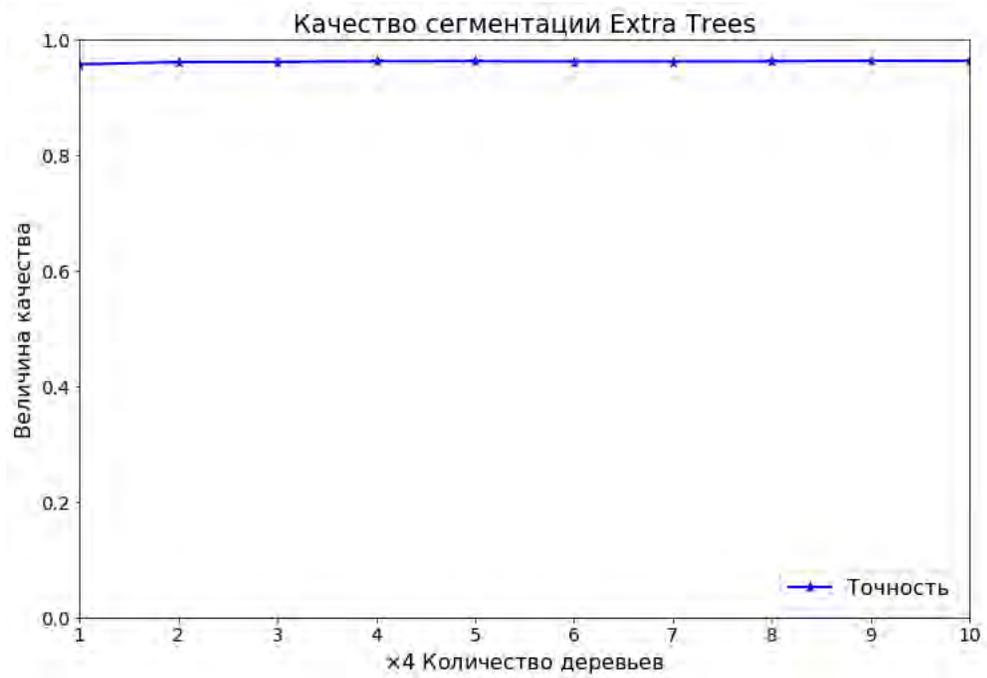


Рис. 30. Поиск оптимального числа деревьев (максимальная глубина дерева равна 18) в случае равномерно сэмплированной выборки

Градиентный бустинг

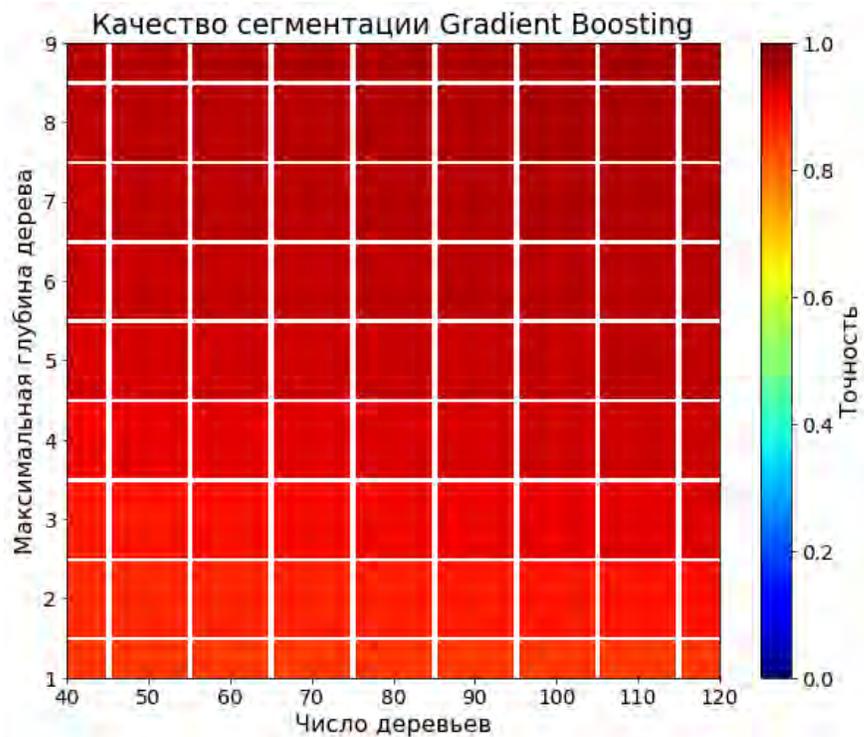


Рис. 31. Поиск оптимальных гиперпараметров алгоритма градиентного бустинга по критерию точности в случае равномерно сэмплированной выборки

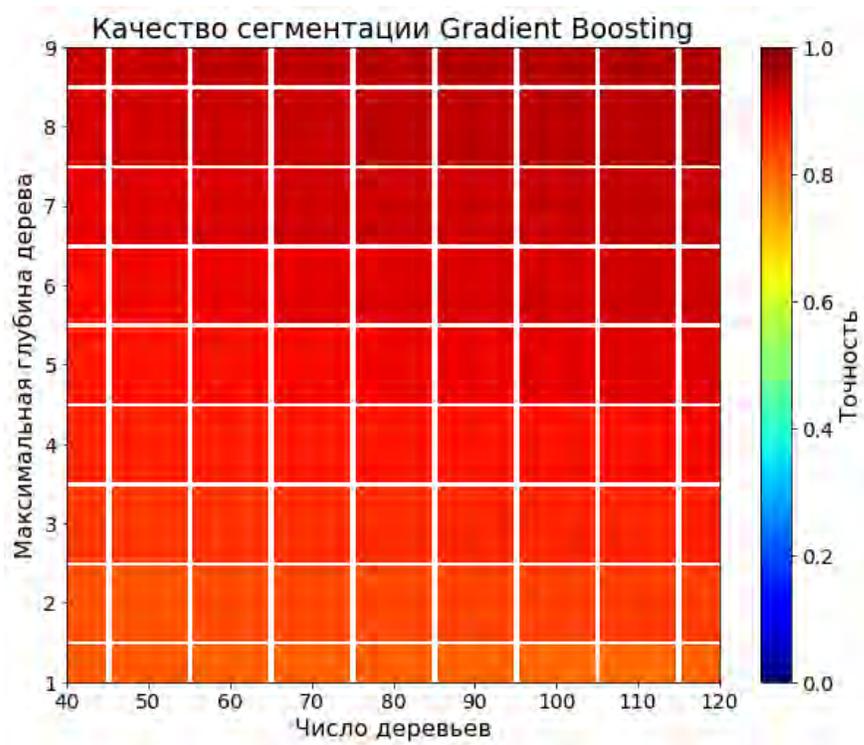


Рис. 32. Поиск оптимальных гиперпараметров алгоритма градиентного бустинга по критерию точности в случае неравномерно сэмплированной выборки

Настройка гиперпараметров в задаче классификации

SVM

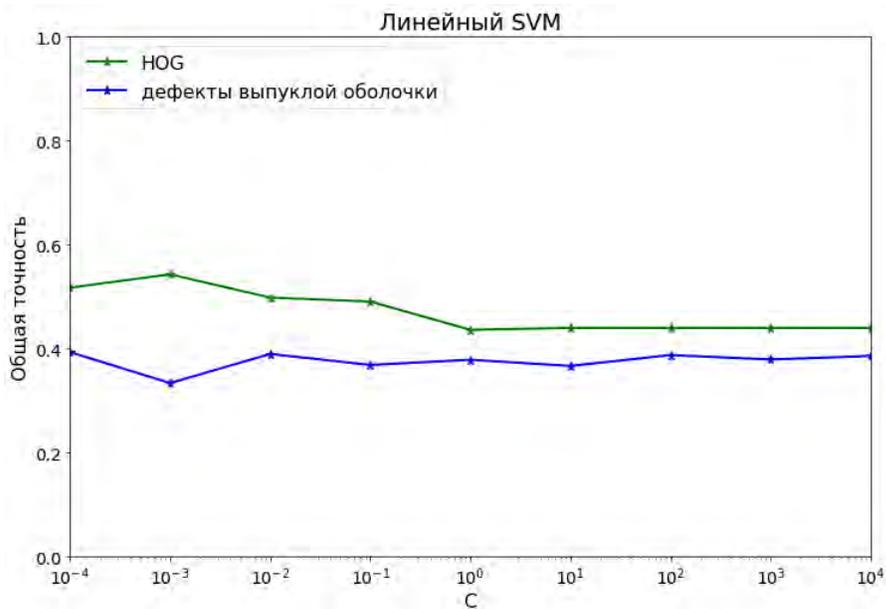


Рис. 33. Зависимость качества линейного SVM от коэффициента регуляризации C

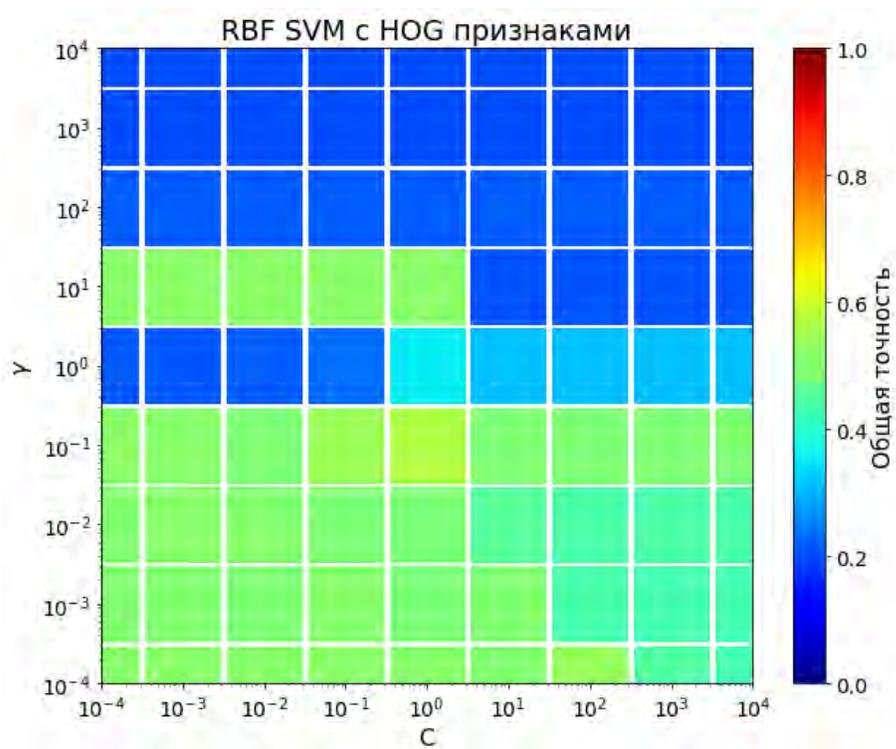


Рис. 34. Зависимость качества SVM с RBF ядром от коэффициента регуляризации (C) и размаха ядра (γ) на HOG признаках

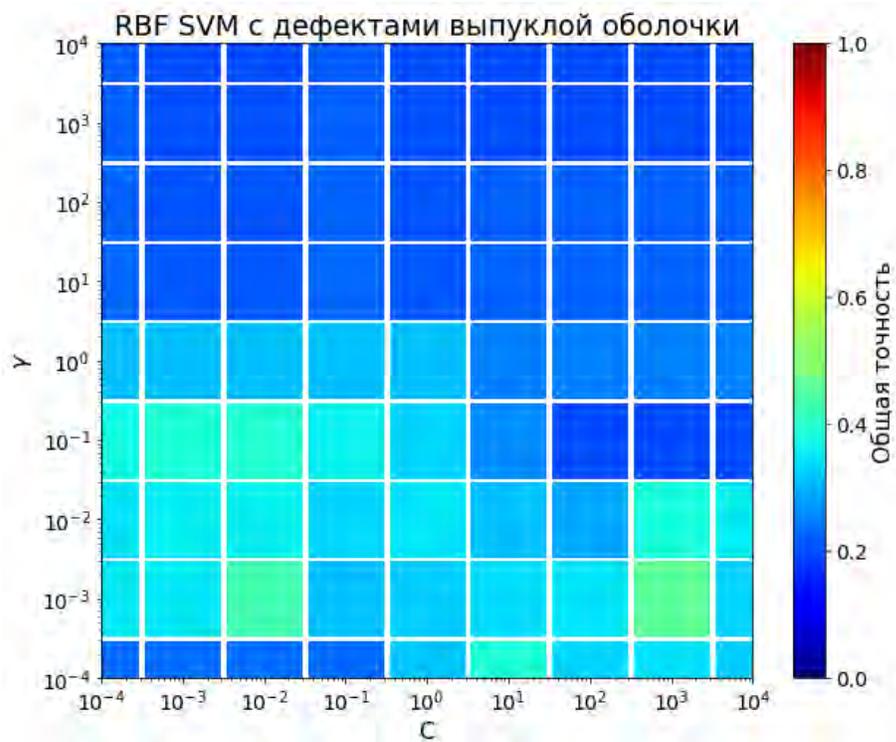


Рис. 35. Зависимость качества SVM с RBF ядром от коэффициента регуляризации (C) и размаха ядра (γ) на дефектах выпуклой оболочки

QDA

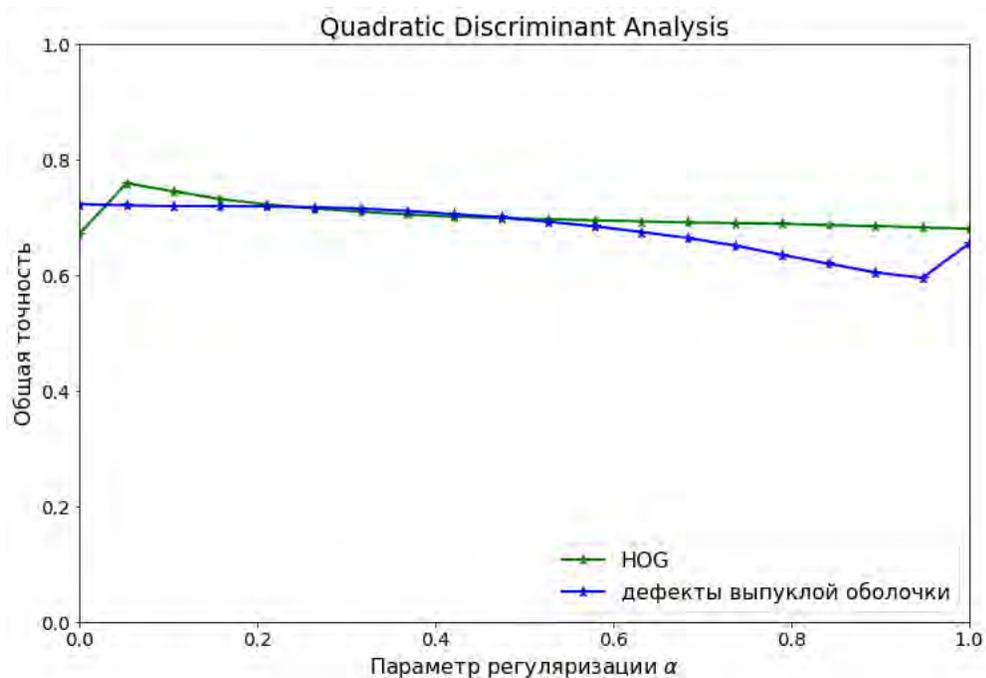


Рис. 36. Зависимость качества QDA от коэффициента регуляризации α для признаковых представлений

Extra Trees

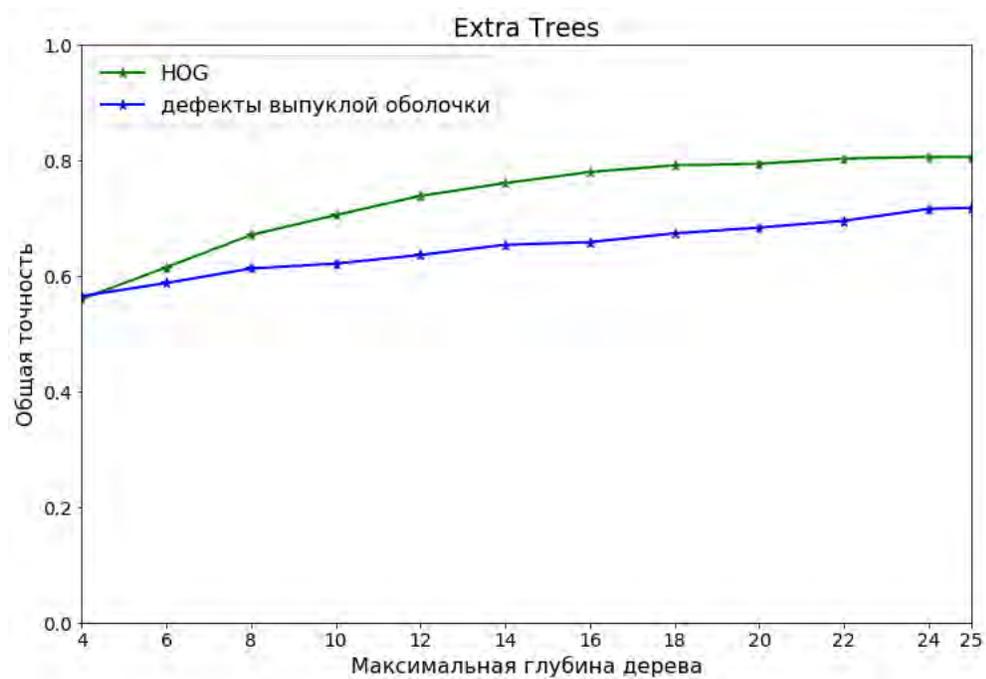


Рис. 37. Зависимость качества Extra Trees от максимальной глубины дерева (количество деревьев равно 10)

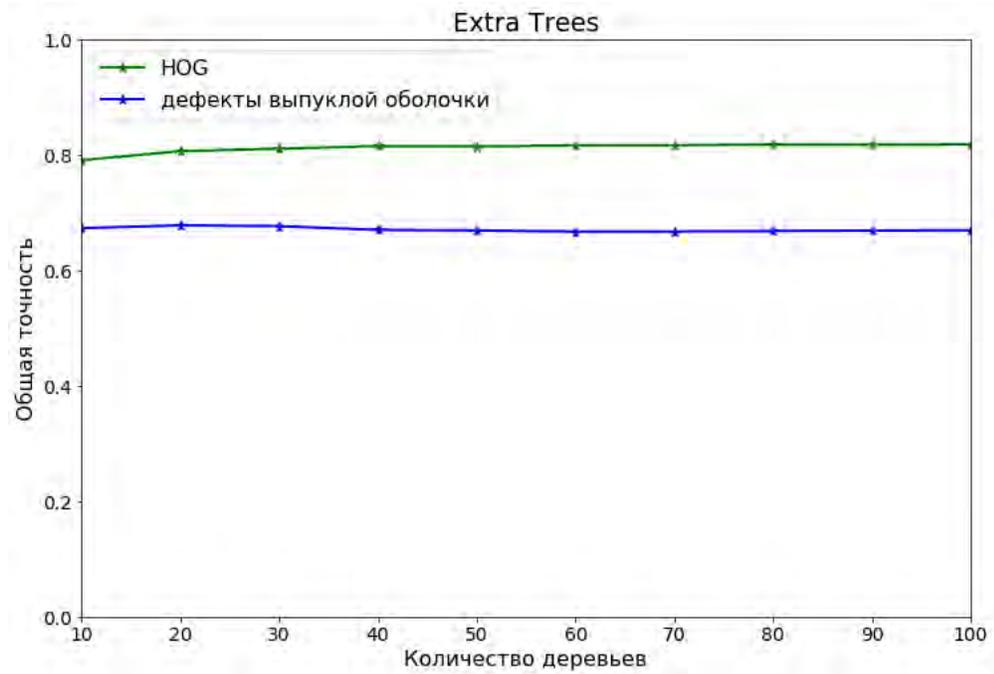


Рис. 38. Зависимость качества Extra Trees от количества деревьев (максимальная глубина дерева равна 18)

Градиентный бустинг

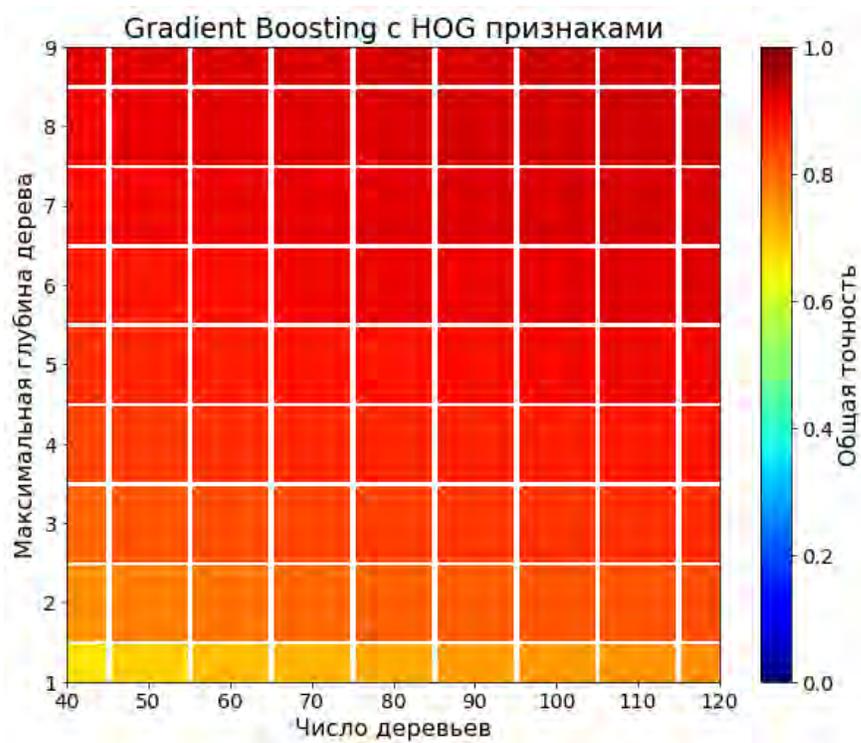


Рис. 39. Зависимость качества градиентного бустинга от гиперпараметров на HOG признаках

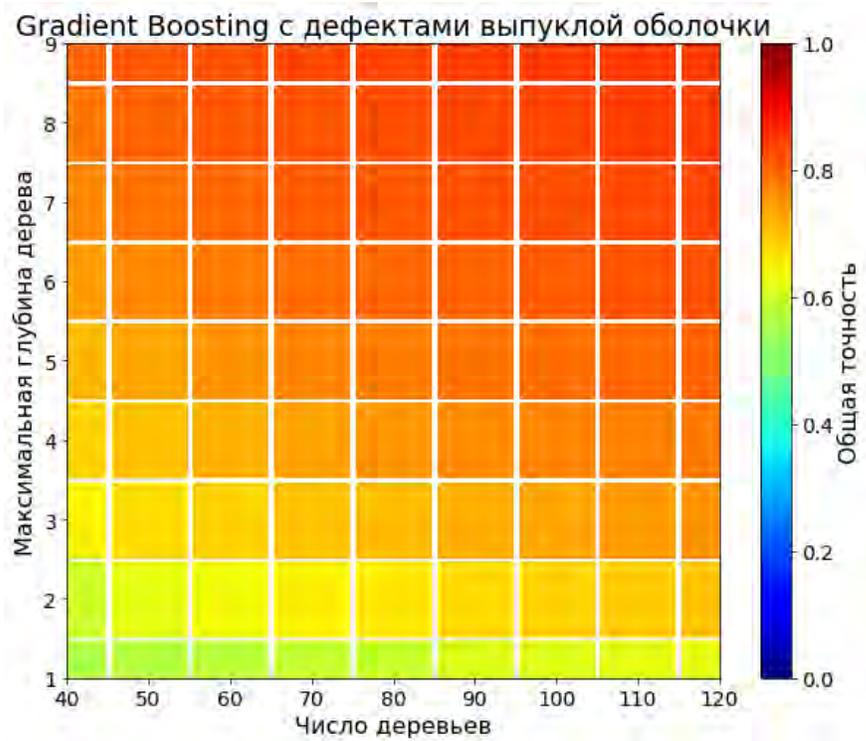


Рис. 40. Зависимость качества градиентного бустинга от гиперпараметров на дефектах выпуклой оболочки