

ОБУЧЕНИЕ КЛАССИФИКАТОРА НА ДЕСКРИПТОРАХ В ПРОСТРАНСТВЕ ПРЕОБРАЗОВАНИЯ РАДОНА¹

© 2020 г. А. Н. Гнеушев^{a,*}, И. А. Матвеев^{a,**}, Н. А. Самсонов^{b,***}

^a ФИЦ ИУ РАН, Москва, Россия

^b МФТИ, Долгопрудный, МО, Россия

*e-mail: gneushev@ccas.ru

**e-mail: matveev@ccas.ru

***e-mail: nikita.samsonov@phystech.edu

Поступила в редакцию 05.12.2019 г.

После доработки 20.12.2019 г.

Принята к публикации 27.01.2020 г.

Задача детектирования объектов на изображении решается путем обучения классификатора на дескрипторах, построенных на основе локального преобразования Радона градиентного поля изображения. Пространство лучевого преобразования Радона рассматривается как аккумуляторное пространство Хафа, в котором строятся проекции. Совокупность локальных проекций формирует дескриптор области изображения объекта, который обобщает известный ранее HOG-дескриптор (histogram of oriented gradients). Исследуются вопросы влияния точности аппроксимации функции вклада в преобразовании Радона, вида локальной нормализации, количества направлений в проекционных гистограммах на результат детектирования пешеходов. Приведены результаты работы предложенного дескриптора в сравнении с HOG-дескриптором и сверточными нейронными сетями на основе архитектур ResNext и MobileNet на базах INRIA, CityScapes.

DOI: 10.31857/S0002338820030051

Введение. Автоматическое детектирование объектов на изображениях и видео — одна из основных задач интеллектуального компьютерного зрения. Цель детектирования состоит в определении наличия и расположения искомого объекта на изображении.

Люди на изображениях являются одним из самых интересных типов объекта для распознавания как с точки зрения сложности решаемой задачи, так и с точки зрения практической значимости. Задача локализации человека на изображениях широко востребована в таких областях, как мониторинг и анализ дорожных ситуаций, обнаружение дорожно-транспортных происшествий, контроль за соблюдением правил дорожного движения, системы безопасности и следящие системы, беспилотные автомобили, робототехника, системы помощи водителю. Сложность распознавания человека на изображении обусловлена несколькими причинами: изменение структуры изображения вследствие движения объекта, неравномерная освещенность сцены, большая вариабельность изображений человека из-за разных ракурсов (поз, размеров, углов поворота), частичные перекрытия фигуры человека другими объектами.

Как правило, множество признаков, которые используются для распознавания объекта, определяются его характерной структурой на изображении. Изображение человека может быть представлено как совокупность силуэтов частей тела, которые являются контурными признаками и представляются на изображении как перепады значений яркости [1]. Однако контуры на изображении могут иметь достаточно сложную и разнообразную структуру, для описания которой необходимо учитывать совокупность контурных признаков — локальных дескрипторов изображения объекта в целом [2–4].

Важнейшим фактором, влияющим на качество детектирования, его устойчивость, является выбранное множество признаков. Чем лучше признаки обладают разделительной способностью

¹ Работа выполнена при финансовой поддержке Минобрнауки России (уникальный идентификатор проекта RFMEFI60719X0312).

стью, тем проще устроено признаковое пространство. В настоящее время эффективным подходом, в котором структура признаков для детектирования объектов формируется автоматически, стало использование многослойных сверточных нейронных сетей глубокого обучения (CNN) [5–8]. В CNN процедура выделения признаков осуществляется в начальных слоях, являющихся частью классификатора, структура признаков формируется в процессе ее обучения и определяется моделью и архитектурой сети. CNN можно рассматривать как обобщенный подход построения признаковой модели объекта, формирующейся на основе только той информации, которая содержится в обучающей выборке. Чем больше “простых” признаков необходимо задействовать для характеристики целевых объектов, тем больше параметров используется для задания CNN, тем она вычислительно сложнее и требовательней к ресурсам и объему обучающей выборки.

Применение небольшого количества специализированных наиболее информативных дескрипторов, получаемых небольшим количеством операций, значительно уменьшает вычислительную сложность системы детектирования, требования к аппаратной платформе и объему обучающей выборки по сравнению с использованием CNN с большой композицией обобщенных “простых” признаков. В работах [9, 10] строится каскад бинарных классификаторов, где признаки, напоминающие базисные вейвлеты Хаара, вычисляются по локальным прямоугольным областям изображения. Однако детекторы, основанные на данном методе, являются неустойчивыми по отношению к таким факторам, как неравномерное изменение освещения и вариабельность изображения объекта. Эффективная схема построения интегрального вектора признаков для задачи детектирования человека на основе локальных контурных дескрипторов – гистограмм ориентаций градиентов в локальных областях изображения используется в методе HOG (histogram of oriented gradients) [11] и подходах, которые его развивают [12–15]. Так, в [12, 13] для построения модели объекта применяется композиция дескрипторов всего объекта на изображении с низким разрешением и дескрипторов частей объекта (руки, ноги, голова) на изображении с высоким разрешением. Такой подход позволяет значительно улучшить качество детектирования, однако с существенным увеличением вычислительной сложности метода. В подходе [15] вместо гистограмм ориентаций градиентов в каждой точке предлагается строить двумерные гистограммы ориентаций градиентов пар точек – гистограммы встречаемости ориентированных градиентов (CoHOG). Такой метод показывает существенный прирост качества описания объектов для детектирования, но при этом сильно увеличивается размерность вектора признаков и количество вычислительных операций.

HOG-дескриптор обладает рядом преимуществ – он устойчив к неравномерному освещению и вычислительно эффективен, однако он недостаточно полно описывает локальные контурные особенности объекта, используя лишь информацию об ориентации, вследствие чего детектор на его основе имеет неудовлетворительный для многих реальных приложений процент ложных срабатываний при принятом уровне пропуска.

Гистограммы ориентаций градиентов можно рассматривать как аппроксимацию проекций модуля градиента на основные направления. В [16, 17] предлагаются методы на основе преобразования Радона для детектирования сложных фигур на изображении. Предлагаемые дескрипторы строятся как проекции в пространстве преобразования Радона бинарных изображений, являются инвариантными относительно поворота, переноса и масштабирования и показывают хорошие результаты, однако для полутоновых изображений результатов получено не было.

В данной работе предлагается подход построения нового семейства локальных дескрипторов объекта на изображении – гистограмм аккумуляторного пространства Хафа (Hough accumulator histograms, HAH) на основе оконного преобразования Хафа. Аккумуляторное пространство получается методом лучевого преобразования Радона градиентного поля изображения. Предложенный подход обобщает HOG-дескриптор, который является частным представителем рассматриваемого семейства. Преобразование Хафа помогает выделять контуры и их пространственные особенности для объектов на изображении, поэтому его применение для построения локальных дескрипторов позволяет учитывать не только локальное распределение ориентаций контурных признаков, но также и их положение. В работе показывается, что задействование дополнительной информации о положении локальных контурных особенностей объекта на изображении позволяет превзойти HOG-дескриптор по качеству. Также, предложенные дескрипторы демонстрируют преимущество по скорости работы на процессорах общего назначения в сравнении с системами на сверточных нейронных сетях, не уступая им по точности.

1. Метод проекций для построения дескриптора в пространстве Радона. Перепады яркости характеризуют контурные границы областей, из которых состоит объект на изображении. Положение, ориентацию и контрастность сегментов границ можно описать в аккумуляторном

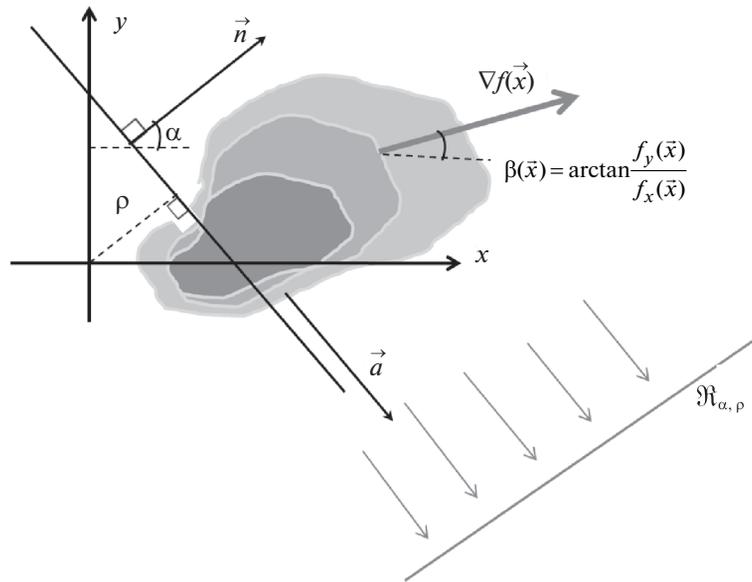


Рис. 1. Преобразование Радона градиентного поля

пространстве Хафа [18]. Совокупность локальных распределений контрастности границ по всем направлениям и положениям будем рассматривать как интегральный дескриптор изображения. Для его построения получим выражение для локального преобразования Хафа путем обобщения лучевого преобразования Радона градиентного поля изображения.

1.1. Преобразование Радона градиентного поля. Будем полагать, что изображение задано некоторой функцией $f(\vec{x}) \in L_2$ с компактным носителем в \mathbb{R}^2 , где $\vec{x} = (x, y)$ – точка изображения. Рассмотрим преобразование Радона функции $f(\vec{x})$ вдоль прямой $L : \vec{x} = \rho \vec{n} + t \vec{a}$, которое определяется выражением [19]

$$\mathcal{R}_{\alpha, \rho} f = \int f(\rho \vec{n} + t \vec{a}) dt, \tag{1.1}$$

где t – свободный параметр прямой L , $\vec{n} = (\cos \alpha, \sin \alpha)$ и $\vec{a} = (-\sin \alpha, \cos \alpha)$ – нормаль и направляющий вектор прямой соответственно, угол α задает ориентацию вектора нормали \vec{n} , параметр $\rho = x \cos \alpha + y \sin \alpha$ – расстояние прямой L до начала системы координат.

Преобразование Радона (1.1) аккумулирует значения функции f вдоль всех прямых L , покрывающих носитель функции f , и характеризует пространственное распределение значений функций по направлениям α на расстояниях ρ от начала системы координат. Таким образом, параметры α и ρ прямых L задают пространство Радона $\mathcal{R}_{\alpha, \rho}$. Преобразование (1.1) обладает свойством обратимости [19], и, таким образом, вся информация о функции f сохраняется.

Пространственные особенности контурных признаков на изображении описываются распределением максимальных изменений функции $f(\vec{x})$, задаваемых ее градиентом $\nabla f(\vec{x})$ [20]. Для описания такого распределения определим лучевое поперечное преобразование Радона поля градиентов функции $f(\vec{x})$ на основе преобразования (1.1) (рис. 1):

$$\mathcal{R}_{\alpha, \rho} \nabla f = \int \langle \nabla f(\rho \vec{n} + t \vec{a}), \vec{n} \rangle dt, \tag{1.2}$$

где $\langle \cdot, \cdot \rangle$ – скалярное произведение, $\nabla f(\vec{x})$ – вектор градиента в точке $\vec{x} = (x, y)$.

Лучевое поперечное преобразование Радона (1.2) для каждой пары параметров α и ρ прямой L аккумулирует значения проекций градиентов функции f на нормаль \vec{n} вдоль соответствующей прямой.

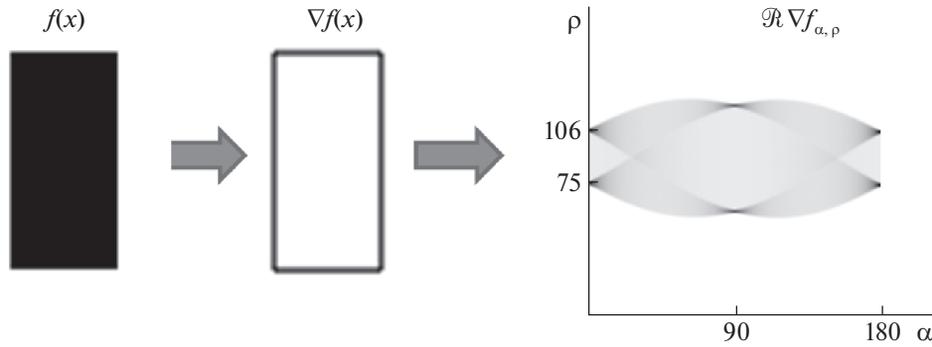


Рис. 2. Локальные экстремумы преобразования Радона градиентного поля изображения прямоугольника характеризуют контурные особенности объекта (изображение инвертировано)

Под знаком интеграла (1.2) стоит производная функции f по направлению \vec{n} в точке \vec{x} . Действительно, полный дифференциал функции f определяется выражением

$$df(\vec{x}) = \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy = \langle \nabla f, d\vec{x} \rangle, \tag{1.3}$$

где приращение $d\vec{x}$ производится вдоль некоторой прямой $\vec{x} = l\vec{n} + r\vec{a}$ с направляющим вектором \vec{n} , нормалью \vec{a} , расстоянием до центра системы координат r и свободным параметром вдоль прямой l . Подставляя приращение $d\vec{x} = \vec{n}dl$ в формулу (1.3), получаем выражение для производной по направлению вектора \vec{n} :

$$\frac{df(\vec{x})}{dl} = \langle \nabla f(\vec{x}), \vec{n} \rangle. \tag{1.4}$$

Таким образом, для каждой точки (α, ρ) в пространстве Радона (1.2) производится интегрирование вдоль прямой L производных функции $f(\vec{x})$ по направлению α на расстоянии ρ от центра системы координат. Такая интерпретация позволяет расширить применение (1.2) с учетом (1.4) на реальные изображения, которые могут быть не достаточно гладкими в окрестности \vec{x} . Наибольшее значение $\mathcal{R}_{\alpha,\rho} \nabla f$ будет соответствовать такому направлению α , вдоль которого преобладают градиенты $\nabla f(\vec{x})$. Таким образом, локальные максимумы в пространстве Радона характеризуют преимущественные направления и локализацию изменений функции f , контурные особенности изображения. На рис. 2 представлено преобразование Радона градиентного поля прямоугольника (изображение инвертировано). В пространстве Радона видны характерные локальные экстремумы, которые являются следствием наибольшего вклада градиента $\nabla f(\vec{x})$ на границах прямоугольника.

Определив угол ориентации вектора градиента ∇f выражением (рис. 1)

$$\beta(\vec{x}) = \text{arctg} \left(\frac{f_y(\vec{x})}{f_x(\vec{x})} \right), \tag{1.5}$$

где $f_x = \partial f / \partial x$, $f_y = \partial f / \partial y$, перепишем формулу (1.2) в виде:

$$\mathcal{R}_{\alpha,\rho} \nabla f = \int \langle \nabla f(\rho\vec{n} + t\vec{a}), \vec{n} \rangle dt = \int |\nabla f(\rho\vec{n} + t\vec{a})| \cos(\alpha - \beta(\vec{x})) dt = \int |\nabla f(\rho\vec{n} + t\vec{a})| q(\alpha, \beta(\vec{x})) dt, \tag{1.6}$$

где $q(\alpha, \beta(\vec{x}))$ – функция, определяющая вклад модуля градиента в точке \vec{x} в направление α , в частном случае равная $q(\alpha, \beta(\vec{x})) = \cos(\alpha - \beta(\vec{x}))$. Выражение (1.6) обобщает лучевое преобразование Радона (1.2) для произвольной функции вклада и, как будет показано далее, позволяет строить различные модели, аппроксимирующие функцию $\cos(\alpha - \beta(\vec{x}))$ и дающие возможность уменьшить количество вычислительных операций.

1.2. Преобразование Хафа градиентного поля изображения. В дискретном случае пространство Радона, полученное с помощью выражения (1.6), можно рассматривать как аккумуляторное пространство Хафа [18]. Для этого зададим разбиение осей пространства Радона $\{\alpha_n\}$ и $\{\rho_m\}$, где $n = \overline{1, N}$ – индексы разбиения по оси α , $m = \overline{1, M}$ – индексы разбиения по

оси ρ . Множество $\{\alpha_n\}$ задает основные направления для учета вклада градиента изображения на множестве $\{\rho_m\}$ расстояний от центра системы координат. Таким образом, аккумуляторное пространство Хафа состоит из $N \times M$ ячеек-аккумуляторов с центрами (α_n, ρ_m) . Для построения аккумуляторного пространства в выражении (1.6) изменим порядок интегрирования. Для каждой ячейки с центром в (α_n, ρ_m) будем суммировать значения подынтегральной функции не вдоль всех прямых, а накапливать результат в них по мере обхода каждой точки $\bar{x} = (x, y)$ градиентного изображения $\nabla f(\bar{x})$ и подсчета вклада градиента в них по всем направлениям разбиения $\{\alpha_n\}$. Таким образом, аккумуляторное пространство Хафа получим из выражения (1.6):

$$\mathcal{H}_{\alpha_n, \rho_m} \nabla f = \sum_{x,y} |\nabla f(\bar{x})| q(\alpha_n, \beta(\bar{x})) p(\rho_m, r_n(\bar{x})), \tag{1.7}$$

$$r_n(\bar{x}) = x \cos \alpha_n + y \sin \alpha_n,$$

где $p(\rho_m, r_n(\bar{x}))$ – интерполяционная функция, определяющая вклад модуля градиента в точке \bar{x} в ячейки по оси ρ и с помощью функции $r_n(\bar{x})$ связывающая точку \bar{x} с ячейкой-аккумулятором (α_n, ρ_m) , в которую следует занести результат суммирования. В простейшем случае, функция $p(\rho_m, r_n(\bar{x}))$ интерполирует вклад модуля градиента в смежные ячейки по оси ρ и имеет вид

$$p(\rho_m, r_n(\bar{x})) = \begin{cases} 1 - \frac{|\rho_m - r_n(\bar{x})|}{\rho_{m+1} - \rho_m}, & \text{при } |\rho_m - r_n(\bar{x})| \leq \rho_{m+1} - \rho_m, \\ 0 & \text{иначе.} \end{cases} \tag{1.8}$$

Построение аккумуляторного пространства (1.7) отличается от классического подхода [18] тем, что модуль градиента для каждой ячейки учитывается с весом, равным его вкладу на данное направление.

Аккумуляторное пространство Хафа для градиентного поля ∇f (1.7) можно рассматривать в качестве пространства, характеризующего контурные признаки изображения, и строить на его основе дескрипторы.

1.3. Оконное преобразование Хафа. Для учета локальных особенностей изображения объекта определим локальное преобразование Хафа с помощью схемы разбиения изображения на ячейки, предложенной в работе [11]. Изображение объекта $f(x, y)$, где x, y – координаты точки изображения размером $X \times Y$, разбивается на смежные квадратные ячейки размером $K \times K$ точек, $I = X/K$ ячеек по горизонтали и $J = Y/K$ ячеек по вертикали. Четыре смежные ячейки объединяются в пересекающиеся квадратные блоки $\{Block_{i,j}\}$, где i, j – индексы блока по вертикали и горизонтали соответственно. Таким образом, каждый блок $Block_{i,j}$ содержит $2K \times 2K$ точек и имеет общие ячейки с соседними блоками. На основе формулы (1.7) определим оконное преобразование Хафа для ячейки с индексом k , где $k = \overline{1, 4}$, внутри блока $Block_{i,j}$ следующим выражением:

$$\mathcal{H}_{\alpha_n, \rho_m}^{i,j,k} \nabla f = \sum_{x,y \in Block_{i,j}} g_k(\bar{x}, \bar{c}_{i,j}) |\nabla f(\bar{x})| q(\alpha_n, \beta(\bar{x})) p(\rho_m, r_n(\bar{x} - o_{i,j}(\bar{x}))), \tag{1.9}$$

где $g_k(\bar{x}, \bar{c}_{i,j})$ – оконная функция, определяющая вес преобразования для каждой точки \bar{x} внутри ячейки k блока $Block_{i,j}$, $\bar{c}_{i,j}$ – координаты центра блока, функция $o_{i,j}(\bar{x})$ задает координату левого нижнего угла ячейки, содержащей точку \bar{x} , и определяет центр локальной системы координат, относительно которой вычисляется значение $r_n(\bar{x})$ при проходе по блоку $Block_{i,j}$.

В качестве функции окна будем использовать композицию функции Гаусса с эффективной шириной $2K$ точек и функции билинейной интерполяции между точками четырех смежных ячеек в виде

$$g_k(\bar{x}, \bar{c}_{i,j}) = \exp\left(-\frac{(\bar{x} - \bar{c}_{i,j})^2}{2\sigma^2}\right) BLI_k(\bar{x} - \bar{c}_{i,j}), \tag{1.10}$$

где σ – среднеквадратичное отклонение функции Гаусса, $BLI_k(\bar{x} - \bar{c}_{i,j})$ – функция билинейной интерполяции между ячейками блока (i, j) для ячейки с индексом k , описанная в работе [11]. Данная оконная функция позволяет строить преобразование Хафа для последовательности пересекающихся блоков изображения методом скользящего окна.

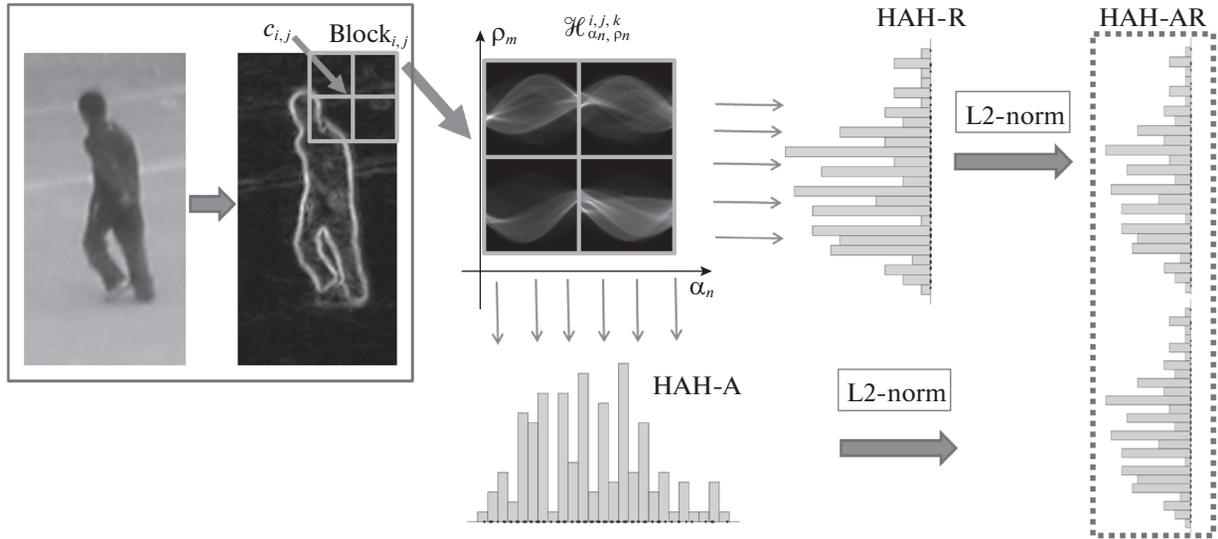


Рис. 3. Построение локальных дескрипторов методом проекций в аккумуляторном пространстве Хафа

1.4. Интегральный дескриптор объекта на изображении. Построим интегральный дескриптор изображения объектов путем объединения множества локальных дескрипторов. Определим локальный дескриптор блока вектором $\vec{b}_{i,j} = \{\vec{u}_{i,j,k}\}_{k=1}^4$, где $\vec{u}_{i,j,k}$ – дескрипторы ячеек в пределах блока.

Используем оконное преобразование Хафа (1.9) для построения локального дескриптора. Для этого уменьшим размерность аккумуляторного пространства путем построения его проекции в пространство одного из параметров (рис. 3). Именно такие проекции называются гистограммами аккумуляторного пространства Хафа (НАН) и используются далее в качестве дескрипторов ячейки $\vec{u}_{i,j,k}$ [21].

Определение 1. Вектор $\vec{u}_{i,j,k}^{НАН-А}$ является НАН-А-дескриптором ячейки и определяется проекцией $\mathcal{H}_{\alpha_n, \rho_m}^{i,j,k}$ (1.9) на подпространство параметра α . Вектор $\vec{b}_{i,j}^{НАН-А}$ – НАН-А-дескриптор блока, объединяющий векторы $\vec{u}_{i,j,k}^{НАН-А}$:

$$\begin{aligned} u_{i,j,k}^n &= \sum_{m=1}^M \mathcal{H}_{\alpha_n, \rho_m}^{i,j,k}, \\ \vec{u}_{i,j,k}^{НАН-А} &= \{u_{i,j,k}^n\}_{n=1}^N, \\ \vec{b}_{i,j}^{НАН-А} &= \{\vec{u}_{i,j,k}^{НАН-А}\}_{k=1}^4. \end{aligned} \quad (1.11)$$

Определение 2. Вектор $\vec{u}_{i,j,k}^{НАН-Р}$ является НАН-Р-дескриптором ячейки и определяется проекцией $\mathcal{H}_{\alpha_n, \rho_m}^{i,j,k}$ (1.9) на подпространство параметра ρ . Вектор $\vec{b}_{i,j}^{НАН-Р}$ – НАН-Р-дескриптор блока, объединяющий векторы $\vec{u}_{i,j,k}^{НАН-Р}$:

$$\begin{aligned} u_{i,j,k}^m &= \sum_{n=1}^N \mathcal{H}_{\alpha_n, \rho_m}^{i,j,k}, \\ \vec{u}_{i,j,k}^{НАН-Р} &= \{u_{i,j,k}^m\}_{m=1}^M, \\ \vec{b}_{i,j}^{НАН-Р} &= \{\vec{u}_{i,j,k}^{НАН-Р}\}_{k=1}^4. \end{aligned} \quad (1.12)$$

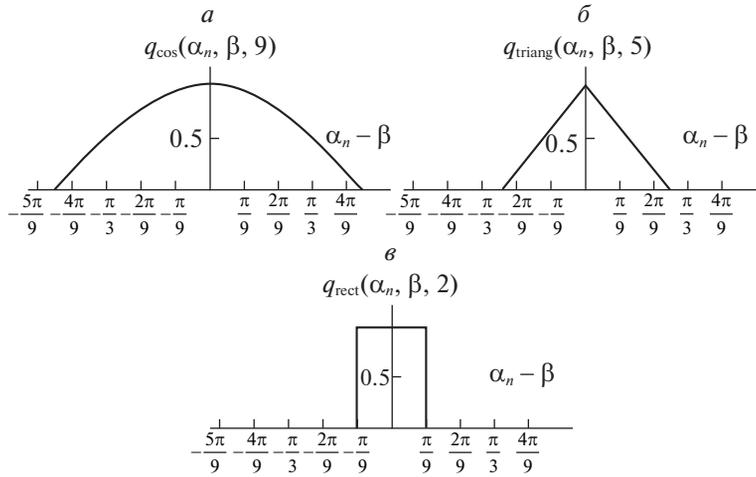


Рис. 4. Модели функций вклада при $N = 9$: а) – $q_{\cos}(\alpha_n, \beta(\bar{x}), l)$ при $l = 9$, б) – $q_{\text{triang}}(\alpha_n, \beta(\bar{x}), l)$ при $l = 5$, в) – $q_{\text{rect}}(\alpha_n, \beta(\bar{x}), l)$ при $l = 2$

О п р е д е л е н и е 3. Вектор $\vec{u}_{i,j,k}^{\text{HAH-AR}}$ является HAH-AR-дескриптором ячейки и объединяет ее HAH-A- и HAH-R-дескрипторы. Вектор $\vec{b}_{i,j,k}^{\text{HAH-AR}}$ – HAH-AR-дескриптор блока, объединяющий векторы $\vec{u}_{i,j,k}^{\text{HAH-AR}}$:

$$\begin{aligned} \vec{u}_{i,j,k}^{\text{HAH-AR}} &= \vec{u}_{i,j,k}^{\text{HAH-A}} \cup \vec{u}_{i,j,k}^{\text{HAH-R}}, \\ \vec{b}_{i,j}^{\text{HAH-AR}} &= \{\vec{u}_{i,j,k}^{\text{HAH-AR}}\}_{k=1}^4. \end{aligned} \tag{1.13}$$

HAH-A-дескриптор использует данные только о распределении ориентаций градиентов по локальной области, HAH-R-дескриптор – только о распределении положений значимых градиентов по расстоянию от края локальной области, HAH-AR-дескриптор сочетает данные обоих типов и, как показали эксперименты, обладает большей информативностью.

HAH-R- и HAH-A-дескрипторы блока нормируются с помощью одной из двух норм, L_2 -нормы –

$$\text{norm}_{L_2}(\vec{b}) = \frac{\vec{b}}{\sqrt{\|\vec{b}\|_2^2 + \varepsilon^2}}, \tag{1.14}$$

либо $(L_2 - \text{hys})$ -нормы –

$$\text{norm}_{L_2-\text{hys}}(\vec{b}) = \text{norm}_{L_2}(\min\{\text{norm}_{L_2}(\vec{b}), \vec{1}h\}), \tag{1.15}$$

где минимум берется по каждой паре компонент векторов, h – пороговое значение, которое введено с целью фильтрации шумовых выбросов в компонентах дескриптора и равно $h = 0.2$, $\vec{1}$ – вектор, каждая компонента которого равна единице [11]. Нормированный HAH-AR-дескриптор блока строится путем объединения нормированных HAH-R- и HAH-A-дескрипторов:

$$\text{norm}(\vec{b}^{\text{HAH-AR}}) = \text{norm}(\vec{b}^{\text{HAH-A}}) \cup \text{norm}(\vec{b}^{\text{HAH-R}}).$$

Интегральный дескриптор изображения объединяет все нормированные дескрипторы блоков (1.16):

$$\vec{v} = \bigcup_i \text{norm}(\vec{b}_{i,j}). \tag{1.16}$$

1.5. А п п р о к с и м а ц и я ф у н к ц и и в к л а д а. Функция $q(\alpha_n, \beta(\bar{x}))$, определяющая вклад модуля градиента в каждое направление, позволяет обобщить преобразование Хафа (1.9). Для одинакового учета вкладов противоположно направленных градиентов в аккумуляторном пространстве предлагается брать модуль косинуса разности между компонентами α_n и направлением градиента [21] (рис. 4, а):

$$q(\alpha_n, \beta(\bar{x})) = |\cos(\alpha_n - \beta(\bar{x}))|. \tag{1.17}$$

Таблица 1. Наименование используемых функций вклада

Функция вклада	Значения параметра l	Название
$q_{\cos}(1.18)$	{2, 5, 9}	cos2, cos5, cos9
$q_{\text{rect}}(1.20)$	{2, 5, 9}	rect2, rect5, rect9
$q_{\text{triang}}(1.19)$	{2, 5, 9}	triang2, triang5, triang9

Такая функция вклада определена для всех компонент разбиения $\{\alpha_n\}$. Аккумуляция всех проекций градиента по полному набору компонент $\{\alpha_n\}$ в преобразовании (1.9), как будет показано далее, требует больше вычислительных операций по сравнению с методом HOG [11, 21]. Более того, определение в каждой точке \bar{x} всех компонент $\{\alpha_n\}$ в формуле (1.9) может оказаться избыточным, так как, во-первых, маленькие по модулю проекции градиента не дают существенный вклад в соответствующую ячейку-аккумулятор, во-вторых, использование только основных, наибольших проекций градиента, может быть достаточным для репрезентативного описания окрестности каждой точки, так как изображения могут быть достаточно гладкими функциями. Уменьшив количество компонент $\{\alpha_n\}$ в каждой точке \bar{x} , можно снизить количество вычислительных операций, сохранив приемлемую точность построения дескрипторов (1.11)–(1.13).

Построим несколько альтернативных моделей функции вклада, аппроксимирующих исходную функцию, но использующих не все компоненты разбиения $\{\alpha_n\}$. Их количество будем ограничивать параметром l . Определим функции вклада q_{\cos} (рис. 4, а), q_{triang} (рис. 4, б) и q_{rect} (рис. 4, в) для всех $n = \overline{1, N}$ с помощью выражений:

$$q_{\cos}(\alpha_n, \beta(\bar{x}), l) = \begin{cases} |\cos(\alpha_n - \beta(\bar{x}))|, & |\alpha_n - \beta(\bar{x})| \leq l \frac{\pi}{2N}, \\ 0 & \text{иначе,} \end{cases} \quad (1.18)$$

$$q_{\text{triang}}(\alpha_n, \beta(\bar{x}), l) = \begin{cases} 1 - \frac{|\alpha_n - \beta(\bar{x})|}{l \frac{\pi}{2N}}, & |\alpha_n - \beta(\bar{x})| \leq l \frac{\pi}{2N}, \\ 0 & \text{иначе,} \end{cases} \quad (1.19)$$

$$q_{\text{rect}}(\alpha_n, \beta(\bar{x}), l) = \begin{cases} 1, & |\alpha_n - \beta(\bar{x})| \leq l \frac{\pi}{2N}, \\ 0 & \text{иначе.} \end{cases} \quad (1.20)$$

Как видно из рис. 4, параметр l определяет эффективную ширину модельной функции. Модельная функция $q_{\cos}(\alpha_n, \beta(\bar{x}), l)$ (1.18) при $l = N$ совпадает с функцией вклада (1.17).

Заметим, что НАН-А-дескриптор с функцией вклада $q_{\text{triang}}(\alpha_n, \beta(\bar{x}), l)$ при $l = 2$ эквивалентен HOG-дескриптору [11, 21]. За счет малой ширины функции и, соответственно, использования всего двух компонент в разбиении $\{\alpha_n\}$ для аккумуляции модуля градиента подсчет HOG-дескриптора производится быстрее по сравнению с альтернативными дескрипторами. Таким образом, оконное преобразование Хафа градиентного поля изображения (1.9) является общим подходом для описания целого семейства HOG-подобных дескрипторов.

Рассматриваемые функции вклада и соответствующие значения параметра l будем кодировать в названии модельной функции следующим образом: <название функции аппроксимации> {значение l }. В табл. 1 приведены используемые названия модельных функций для разных значений параметра l .

1.6. Детектор объекта в пространстве признаков на основе обучения. Построение детектора объекта включает две подзадачи: построение дескриптора области изображения в пространстве признаков (1.16) и его использование для бинарной классификации, отнесения к классу изображений целевого объекта или к классу фона (рис. 5). Как и в работе [11], будем использовать линейный классификатор и обучать его методом опорных векторов (support vector machine, SVM) с линейным ядром.

Веса классификатора, определяющие разделяющую гиперплоскость в пространстве признаков (1.16), находятся из процедуры обучения по обучающей выборке из базы изображений,

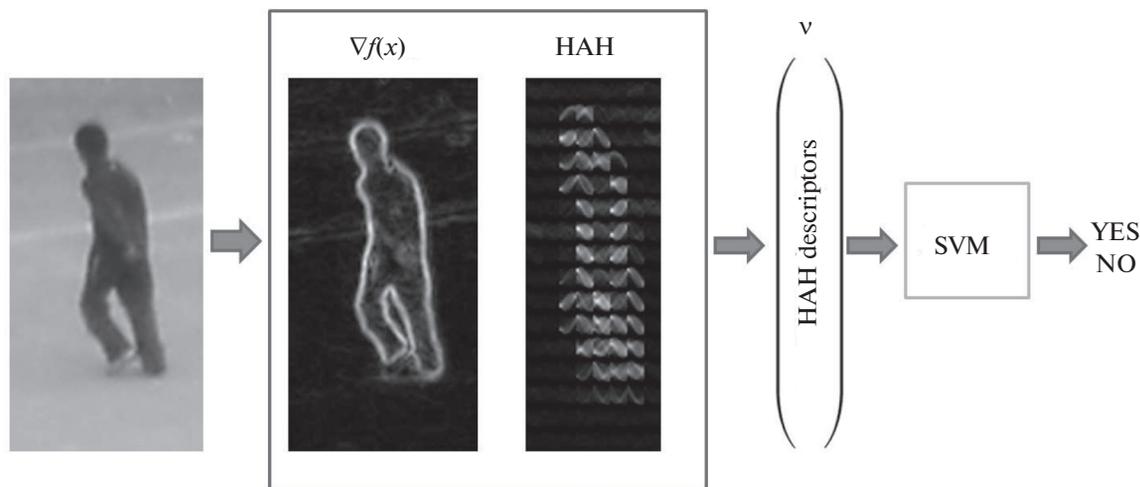


Рис. 5. Схема построения детектора пешехода на основе НАН-дескрипторов и SVM-классификатора

содержащий два подмножества изображений: с положительными примерами, содержащими пешеходов, и отрицательными примерами, содержащими фон.

Как и в [11], процесс обучения будем проводить в два этапа. На первом этапе обучение производится на выборке, составленной из положительных примеров (пешеходов) и случайной выборки областей из изображений фона (отрицательные примеры). После первого этапа обученный классификатор используется для сбора “сложных” областей фона, где детектор ошибается и ложно срабатывает. Эти примеры добавляются в обучающую выборку в качестве отрицательных. На втором этапе классификатор заново обучается по расширенной выборке.

2. Вычислительный эксперимент. Для обучения и тестирования детекторов на основе разработанных методов использовались базы изображений пешеходов INRIA [22] и CityPersons [23, 24].

Обучающая база INRIA содержит 2478 цветных изображений пешеходов размером 160 на 96 точек. Для обучения из каждого изображения выбиралось центральное окно размером 128 на 64 точки. Обучающая выборка содержит 1218 изображений фона. Для обучения в качестве отрицательных примеров выбираются 12180 окон размером 128 на 64 точки (по 10 случайных окон с каждого изображения). Тестовая база изображений INRIA содержит 1132 изображения пешеходов размером 134 на 70 точки. Для теста из 453 изображений фона с помощью скользящего окна с шагом 4 точки по горизонтали и вертикали выбирались 3526548 окон размером 128 на 64 точки.

Обучающая база CityPersons содержит 2975 и 500 цветных изображений дорожных сцен в обучающей и тестовых выборках соответственно. При этом на обучающей размечено 19654, а на тестовой 3938 пешеходов. Так как в работе ставилась задача разработать дескриптор объекта и не исследовались способы модификации классификатора с целью детектирования загороженных пешеходов, для обучения и тестирования на изображениях выбирались области, содержащие пешеходов, в которых площадь перекрытия другими объектами составляет не более 20% от площади самого пешехода. Таким образом, вырезанных областей, содержащих пешеходов, получилось 10170 и 1069 в обучающей и тестовой выборках соответственно. Как и для изображений из базы INRIA, вырезанные области имеют размер 160 на 96 точек для обучающей и 134 на 70 точек для тестовой выборки. Отрицательные примеры выбирались из базы INRIA, при этом для обучения выбирались по 20 случайных окон размером 128 на 64 точки из каждого изображения фона.

Для оконного преобразования Хафа (1.9) цветного изображения градиент изображения $\nabla f(\vec{x})$ выбирается из трех градиентов цветовых каналов; берется градиент с максимальным модулем:

$$\nabla f(\vec{x}) = \nabla f_{c^*}(\vec{x}), \quad c^* = \arg \max_{c \in \{R, G, B\}} |\nabla f_c(\vec{x})|, \quad (2.1)$$

где $f_c(\vec{x})$ – монохромное изображение, соответствующее цветовому каналу c изображения $f(\vec{x})$. Для получения градиентов цветовых каналов частные производные (1.3) рассчитывались с помощью простой разности смежных точек в вертикальном и горизонтальном направлениях.

Для измерения точности обучаемых детекторов строились DET-кривые (detection error tradeoff) на тестовой выборке изображений. В логарифмическом масштабе по оси абсцисс откла-

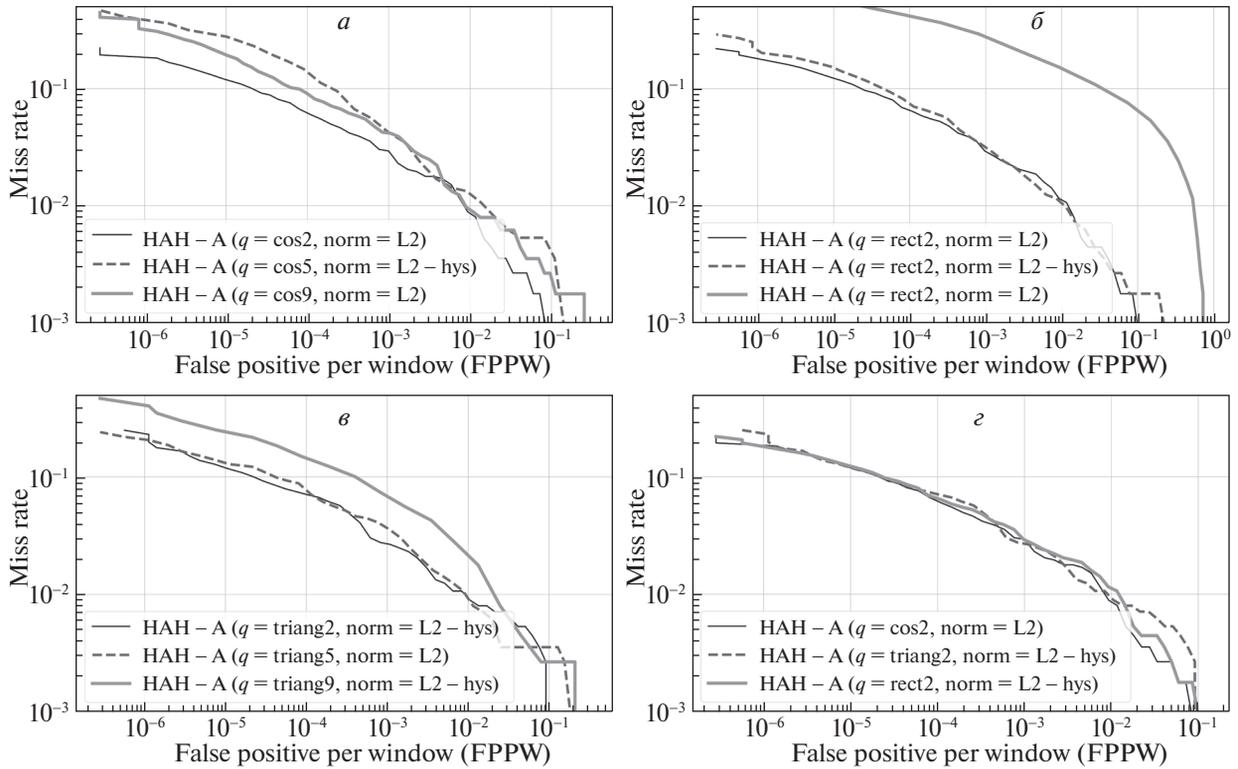


Рис. 6. DET-кривые детектора на HAH-A-дескрипторе с различными моделями функции вклада и типом нормализации: а) – q_{cos} , б) – q_{rect} , в) – q_{triag} , г) – сравнение лучших моделей функции вклада между собой

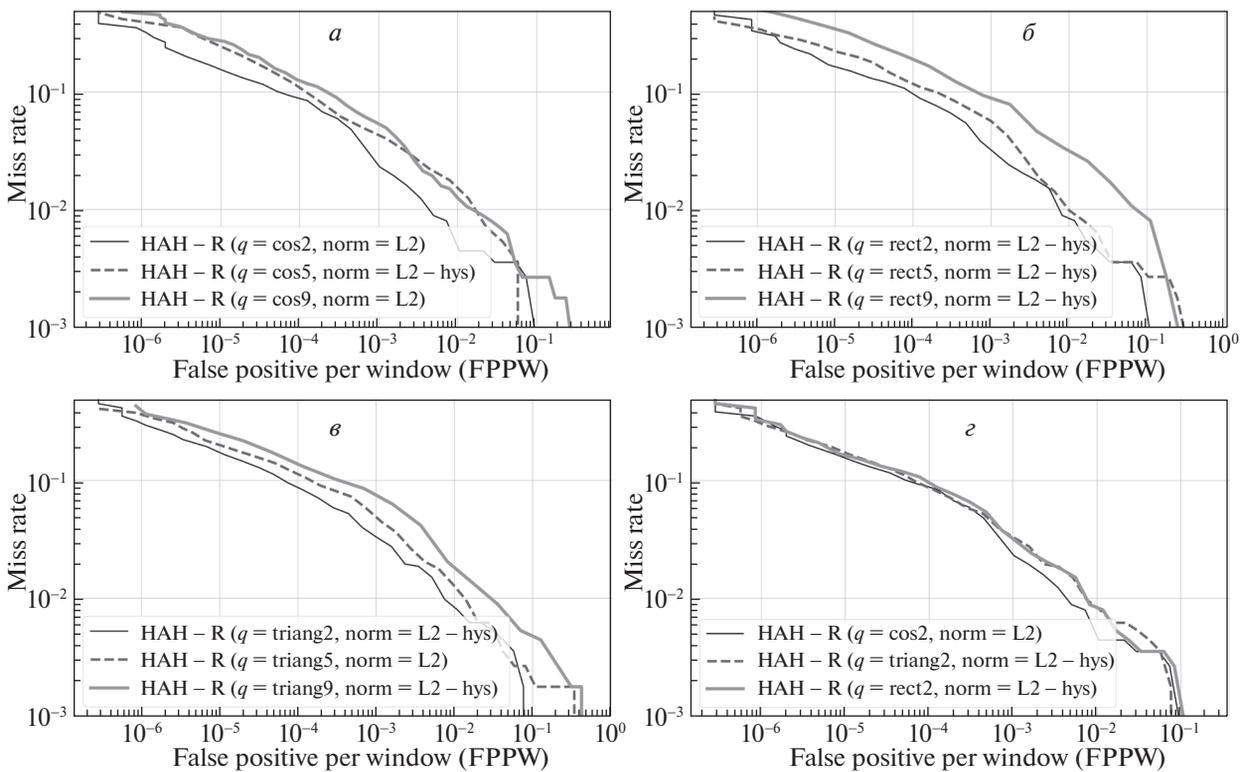


Рис. 7. DET-кривые детектора на HAH-R-дескрипторе с различными моделями функции вклада и типом нормализации: а) – q_{cos} , б) – q_{rect} , в) – q_{triag} , г) – сравнение лучших моделей функции вклада между собой

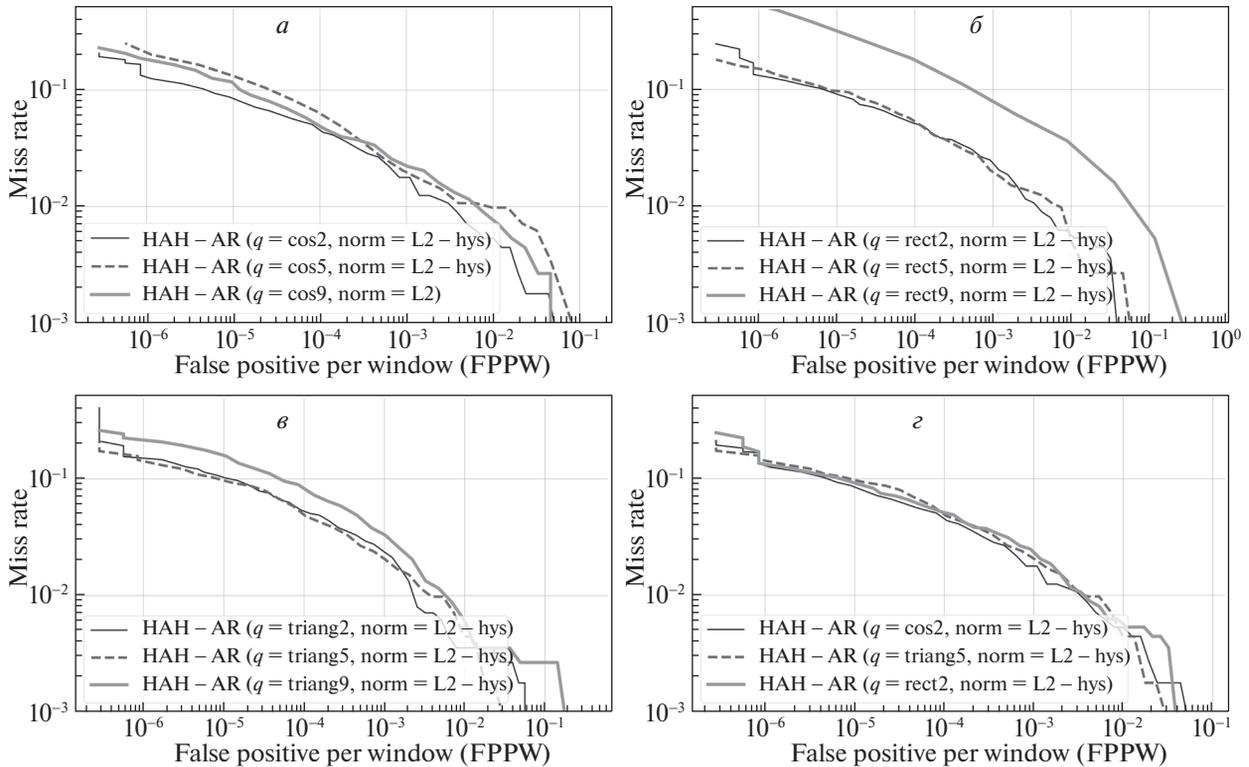


Рис. 8. DET-кривые детектора на HAH-AR-дескрипторе с различными моделями функции вклада и типом нормализации: а) – q_{cos} , б) – q_{rect} , в) – q_{triang} , г) – сравнение лучших моделей функции вклада между собой

дывалась доля ложных обнаружений объекта (FalsePositive) на негативных примерах (TotalNegative), FPPW (false positive per window), по оси ординат откладывалась доля неверно отвергнутых примеров (FalseNegative) на положительной выборке (TotalPositive), MissRate, которые определяются выражениями:

$$FPPW = \frac{FalsePositive}{TotalNegative}, \quad MissRate = \frac{FalseNegative}{TotalPositive}. \quad (2.2)$$

При построении кривой на тестовых примерах, содержащих пешехода, выбиралось окно с наибольшим откликом классификатора среди всех окон в окрестности радиусом два пиксела относительно центрального окна.

Все вычислительные эксперименты и замеры времени работы алгоритмов проводились на компьютере с процессором Intel(R) Core(TM) i7-950 3.07 ГГц с 9 ГБ оперативной памяти, графическим ускорителем Nvidia GTX 1080 1607 МГц в системе Ubuntu 16.04. Замеры времени работы на CPU производились при запуске алгоритмов в однопоточном режиме. Детекторы на основе дескрипторов HAH и HOG тестировались только на CPU.

2.1. Выбор оптимальных параметров для HAH-дескрипторов. Для выбора оптимальных параметров предложенных HAH-дескрипторов детектор пешехода (рис. 5) обучался и тестировался на изображениях из базы INRIA по описанной методике.

Проведенные эксперименты подтвердили результаты работы [11], что для оконного преобразования Хафа (1.9) изображения с размером $X = 64$ и $Y = 128$ и размером ячейки $K = 8$ оптимальное количество компонент N разбиения углов $\{\alpha_n\}$ равно 9. Этот параметр был зафиксирован во всех дальнейших экспериментах.

Для выбора наилучшей модели функции вклада (1.18)–(1.20) и значения параметра l классификатор SVM обучался для каждого дескриптора (1.11)–(1.13) с различными типами нормализации (1.14)–(1.15). Значения параметра l выбирались из множества $\{2, 5, 9\}$.

На рис. 6–8 приведены DET-кривые детекторов на основе HAH-A-, HAH-R-, HAH-AR-дескрипторов соответственно, обученных с различными моделями функции вклада. Для каждой

Таблица 2. Сравнение точности детекторов на НАН-дескрипторах с различными моделями функции вклада с лучшим параметром l

Функция вклада	MissRate (при FPPW = 10^{-4})
Детектор на НАН-А-дескрипторе	
cos2	0.062
rect2	0.065
triang2	0.07
Детектор на НАН-R-дескрипторе	
cos2	0.088
rect2	0.098
triang2	0.088
Детектор на НАН-AR-дескрипторе	
cos2	0.045
rect2	0.05
triang5	0.05

Таблица 3. Сравнение точности и скорости детекторов на НАН-AR-дескрипторе с различными моделями функции вклада и количеством компонент M разбиения по оси ρ

Параметры НАН-AR	MissRate (при FPPW = 10^{-4})	Время (CPU), мс
cos9, $M = 16$	0.05	38
cos2, $M = 16$	0.045	16
rect2, $M = 16$	0.05	11
triang5, $M = 20$	0.048	31

функции вклада были отобраны детекторы с наилучшим значением параметра l . На рис. 6, 7, 8, 9 даны кривые для отобранных детекторов, показатели точности которых приведены в табл. 2.

Из этих результатов видно, что для каждого дескриптора, оптимального по l , влияние функции вклада на точность мало. Лучший результат для всех дескрипторов получается при минимальной эффективной ширине модели $l = 2$. Для НАН-AR-дескриптора точность детектора для $l = 2$ и для $l = 5$ не отличается, при этом время работы детектора с $l = 2$ в 2 раза меньше, чем с $l = 5$. Таким образом, использование только основных, наибольших проекций градиента в оконном преобразовании Хафа (1.9) достаточно для точности работы детектора и позволяет сократить количество вычислительных операций, уменьшив время работы детектора более чем в 2 раза (табл. 3).

Из табл. 2 видно, что детектор на НАН-AR-дескрипторе обладает наибольшей точностью (рис. 9). Он показывает почти в 2 раза меньшую ошибку, чем детектор на НАН-R-дескрипторе, и в 1.4 раза меньшую ошибку, чем детектор на НАН-А-дескрипторе. Таким образом, использование информации не только о локальных распределениях ориентаций градиентов изображения, но и информацию о распределении их положений позволяет значительно улучшить качество детектирования объектов.

Были проведены эксперименты для выбора оптимального значения количества компонент M разбиения $\{\rho_m\}$ аккумуляторного пространства Хафа для детектора на НАН-AR-дескрипторе. Значения для параметра M выбирались из множества $\{4, 8, 12, 16, 20\}$. На рис. 10 приведены DET-кривые детекторов, обученных с выбранными моделями функции вклада и различными значениями параметра M . Для каждой функции вклада были отобраны детекторы с наилучшим значением параметра M . На рис. 10, 2 показаны кривые, отобранных детекторов, показатели точности и времени работы которых приведены в табл. 3.

Из табл. 3 видно, что использование моделей функций вклада с минимальной характерной шириной l позволяет получить точность детектора не хуже, чем с исходной функцией вклада (1.17) для всех N компонент разбиения $\{\alpha_n\}$, при этом достигается сокращение времени детектирования

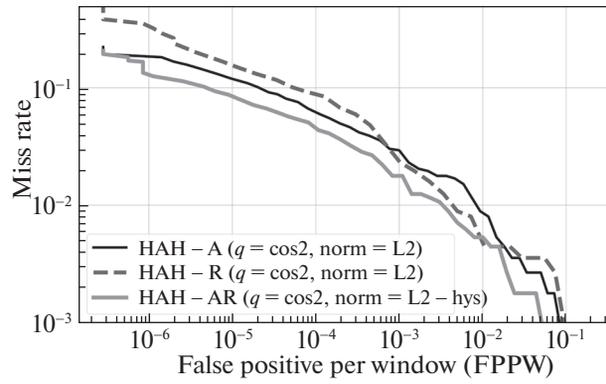


Рис. 9. DET-кривые детектора на основе HAH-A-, HAH-R- и HAH-AR-дескрипторов с наилучшими моделями функции вклада и типов нормализации

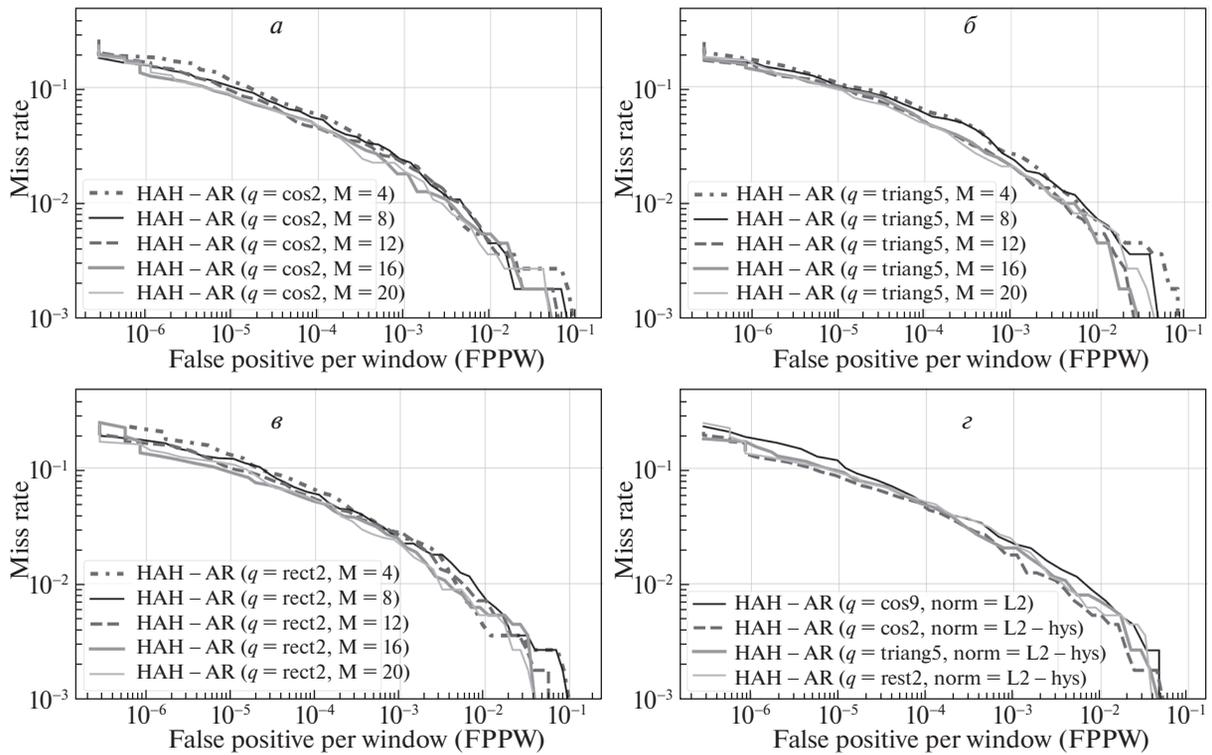


Рис. 10. DET-кривые детектора на HAH-AR-дескрипторе с наилучшими моделями функции вклада и различным количеством компонент M по оси ρ : а) – $\cos 2$, б) – $\text{triang} 5$, в) – $\text{rect} 2$, г) – сравнение детекторов с наилучшими параметрами и с функцией вклада $\cos 9$

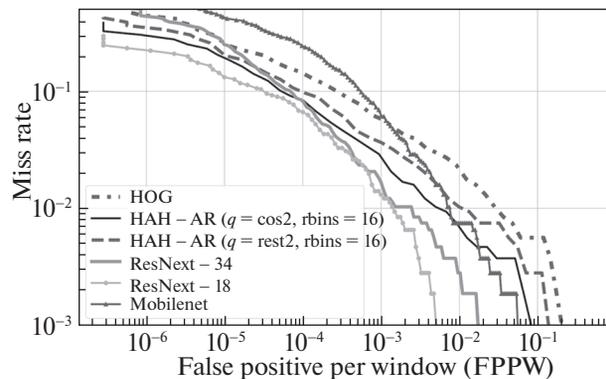


Рис. 11. Сравнение DET-кривых детекторов на основе предлагаемых дескрипторов и нейронных сетей

Таблица 4. Сравнение точности и скорости работы детекторов

Метод	MissRate (при FPPW = 10^{-4})		Время, мс	
	INRIA	CityPersons	CPU	GPU
HOG	0.13	0.15	5.5	—
НАН-AR (cos2)	0.045	0.08	16	—
НАН-AR (rect2)	0.05	0.10	11	—
Mobilenet	—	0.25	20	2.4
ResNext-18	—	0.07	42	2.5
ResNext-34	—	0.08	81	3.8

ния более чем в 2 раза. По результатам экспериментов наиболее эффективными дескрипторами являются НАН-AR с функциями вклада \cos^2 и rect^2 .

2.2. Сравнение с существующими методами. Предлагаемые в работе детекторы сравнивались с детектором на основе HOG-дескриптора [11], а также с детекторами на основе сверточных нейронных сетей архитектур ResNext [25] и Mobilenet [26]. Для сравнения с нейросетевыми детекторами для обучения и тестирования использовалась база CityPersons, так как нейросети для обучения требуют более объемной базы, чем INRIA. Детектор на основе дескрипторов обучался по описанной методике в два этапа. Детектор на основе сверточных нейронных сетей обучался в один этап.

На рис. 11 представлены DET-кривые для сравниваемых детекторов, полученных для базы CityPersons. В табл. 4 приведены показатели точности и времени работы детекторов.

Результаты экспериментов показывают, что использование НАН-AR-дескриптора позволяет получить детектор с ошибкой в 2 раза меньшей, чем детектор на HOG-дескрипторе, уступая по времени работы в 2 раза. Детектор на НАН-AR-дескрипторе близок к точности нейросетевого детектора архитектуры ResNext-18, при этом обгоняет его по времени более чем в 2.5 раза.

Заключение. Предложен новый подход построения дескриптора изображения объектов для задачи обнаружения пешеходов, обобщающий метод HOG. На основе лучевого поперечного преобразования Радона градиентного поля изображения предложено оконное преобразование Хафа градиентного изображения для построения локального аккумуляторного пространства, в котором модули градиента накапливаются с учетом их вклада в различные направления. Проекция в этом пространстве позволяет построить дескрипторы областей изображения. Предложенный подход обосновывает HOG-дескриптор, который является частным случаем построенного НАН-AR-дескриптора с узкой моделью функцией вклада градиента. Использование только наибольших проекций градиента не ухудшает точность построенного на дескрипторе детектора и при этом значительно сокращает количество вычислительных операций. Сравнение детектора НАН-AR с другими детекторами показывает его преимущество по скорости работы на процессоре общего назначения при конкурентной точности.

СПИСОК ЛИТЕРАТУРЫ

1. Себряков Г.Г., Сошников В.Н., Кикин И.С., Ишутин А.А. Алгоритм распознавания объектов на оптико-электронных изображениях наземных сцен, основанный на оценке ковариационных матриц градиентных функций поля яркости // Вестник компьютерных и информационных технологий. 2013. V. 109. № 7. С. 14–19.
2. Визильтер Ю.В., Желтов С.Ю. Использование проективных морфологий в задачах обнаружения и идентификации объектов на изображениях // Изв. РАН. ТиСУ. 2009. № 2. С. 125–138.
3. Гнеушев А.Н. Построение и оптимизация текстурно-геометрической модели изображения лица в пространстве базисных функций Габор // Изв. РАН. ТиСУ. 2007. № 3. С. 85–96.
4. Гнеушев А.Н. Оптимизация текстурно-геометрической модели изображения для оценивания параметров лица // АиТ. 2012. № 1. С. 159–168.
5. Zhang L., Lin L., Liang X., He K. Is Faster R-CNN Doing Well for Pedestrian Detection? // Proc. 14th European Conf. Computer Vision. Amsterdam, Netherlands, 2016. P. 443–457.
6. Cai Z., Fan Q., Feris R.S., Vasconcelos N. A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection // Proc. 14th European Conf. Computer Vision. Amsterdam, Netherlands, 2016. P. 354–370.

7. *Li J., Liang X., Shen S.* Scale-aware Fast R-CNN for Pedestrian Detection // IEEE Trans. Multimedia. 2017. V. 20. № 4. P. 985–996.
8. *Du X., El-Khamy M., Lee J., Davis L.S.* Fused DNN: a Deep Neural Network Fusion Approach to Fast and Robust Pedestrian Detection // Proc. 2017 IEEE Winter Conf. Applications of Computer Vision (WACV). Santa Rosa, CA, USA, 2017. P. 953–961.
9. *Viola P., Jones M.J.* Rapid Object Detection Using a Boosted Cascade of Simple Features // Proc. 2001 IEEE Conf. Computer Vision and Pattern Recognition. Kauai, HI, USA, 2001. V. 1. P. 511–518.
10. *Viola P., Jones M.J., Snow D.* Detecting Pedestrians Using Patterns of Motion and Appearance // 9th Int. Conf. Computer Vision. Nice, France, 2003. V. 1. P. 734–741.
11. *Dalal N., Triggs B.* Histograms of oriented gradients for human detection // 2005 IEEE Conf. Computer Vision and Pattern Recognition. San Diego, CA, USA, 2005. V. 1. P. 886–893.
12. *Felzenszwalb P., McAllester D., Ramanan D.* A Discriminatively Trained, Multiscale, Deformable Part Model // Proc. 2008 IEEE Conf. Computer Vision and Pattern Recognition. Anchorage, AK, USA, 2008. P. 1–8.
13. *Felzenszwalb P.F.B., Girshick R., McAllester D., Ramanan D.* Object Detection with Discriminatively Trained Part Based Models // IEEE Trans. Pattern Analysis and Machine Intelligence. 2010. V. 32. № 9. P. 1627–1645.
14. *Sun D., Watanada J.* Detecting Pedestrians and Vehicles in Traffic Scene Based on Boosted HOG Features and SVM // IEEE 9th Int. Symp. Intelligent Signal Processing. Siena, Italy, 2015. P. 1–4.
15. *Watanabe T., Ito S., Yokoi K.* Co-occurrence Histograms of Oriented Gradients for Pedestrian Detection // Advances in Image and Video Technology. 2009. V. 5414. P. 37–47.
16. *Tabbone S., Wendling L., Salmon J.P.* A New Shape Descriptor Defined on the Radon Transform // Computer Vision and Image Understanding. 2006. V. 102. 1. P. 42–51.
17. *Nacereddine N., Tabbone S., Ziou D., Hamami L.* Shape-based Image Retrieval Using a New Descriptor Based on the Radon and Wavelet Transforms // 20th Int. Conf. Pattern Recognition. Istanbul, Turkey, 2010. P. 1997–2000.
18. *Ballard H.* Generalizing the Hough Transform to detect arbitrary Shapes // Pattern Recognition. 1981. V. 13. № 2. P. 111–122.
19. *Светов И.Е.* Формулы обращения для восстановления двумерных гармонических векторных полей по известным лучевым преобразованиям // Сибирские электронные математические известия. 2015. Т. 12. С. 436–446.
20. *Гнеушев А.Н., Мурынин А.Б.* Адаптивный градиентный метод выделения контурных признаков объектов на изображениях реальных сцен // Изв. РАН. ТиСУ. 2003. № 6. С. 128–135.
21. *Гнеушев А.Н., Самсонов Н.А.* Дескриптор в аккумуляторном пространстве Хафа градиентного поля изображения для детектирования пешеходов // Машинное обучение и анализ данных. 2017. Т. 3. № 3. С. 203–215.
22. INRIA Person Dataset. Available at: <http://pascal.inrialpes.fr/data/human/> (дата обращения 2017-06-04).
23. *Zhang S., Benenson R., Schiele B.* CityPersons: A Diverse Dataset for Pedestrian Detection // 2017 IEEE Conf. Computer Vision and Pattern Recognition. Honolulu, HI, USA, 2017. P. 4457–4465.
24. *Cordts M., Omran M., Ramos S., Rehfeld T., Enzweiler M., Benenson R., Franke U., Roth S., Schiele B.* The Cityscapes Dataset for Semantic Urban Scene Understanding // 2016 IEEE Conf. Computer Vision and Pattern Recognition. Las Vegas, NV, USA, 2016. P. 3213–3223.
25. *He K., Zhang X., Ren S., Sun J.* Deep Residual Learning for Image Recognition // 2016 IEEE Conf. Computer Vision and Pattern Recognition. Las Vegas, NV, USA, 2016. P. 770–778.
26. *Howard A.G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H.* MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications // Available at: <https://arxiv.org/abs/1704.04861> (дата обращения 2019-06-23).