

## ОПИСАНИЕ ДЕСКРИПТОРА, ОСУЩЕСТВЛЯЮЩЕГО КАТЕГОРИЗАЦИЮ ИЗОБРАЖЕНИЙ С УЧЕТОМ ГЕОМЕТРИЧЕСКИХ СТРУКТУР ПО ВСЕМ ЦВЕТАМ

Ляхов А. Ф.<sup>1</sup>, Чнегов Е. И.<sup>1</sup>

<sup>1</sup>Национальный исследовательский Нижегородский государственный университет им. Н. И. Лобачевского, Нижний Новгород, Россия

### Аннотация

Предлагается новый оригинальный дескриптор, позволяющий осуществлять категоризацию, как статичных изображений, так и видеопотока. Дескриптор, учитывает цветовую гамму изображения и геометрическую структуру по всем цветам, то есть фрактальность изображения. Проводится сравнение эффективности дескриптора с описанием дескрипторов в литературе.

**Ключевые слова:** дескриптор, распознавание образов, классификация изображений.

**Цитирование:** Ляхов А. Ф., Чнегов Е. И. Описание дескриптора осуществляющего категоризацию изображений с учетом геометрических структур по всем цветам // Компьютерные инструменты в образовании. 2017. № 6. С. 31–43.

### 1. ВВЕДЕНИЕ

Теория и практика распознавания образов является одним из важнейших направлений современной прикладной математики и IT-технологий. В основе теории распознавания образов лежат методы классификации и идентификации объектов на основе математической статистики и теории вероятностей [1–6]. Среди задач, решаемых при распознавании образа, важное место занимают задачи об определении наличия объекта на изучаемом изображении [7–9]. Задачи подобного типа являются задачами категоризации изображения.

Один из методов решения задачи категоризации основывается на введении понятия дескриптора изображения [11, 13–15]. Дескриптор — это объект, содержащий информацию о визуальных составляющих изображения, позволяющих осуществить распознавание объекта на изображении. К полезной информации об изображении можно отнести цвет, форму, текстуру. Дескрипторы изображения принято разделять на два типа:

- 1) содержащие общую информацию об изображении (**General information descriptors**),
- 2) содержащие информацию о конкретных объектах на изображении (**Specific domain information descriptors**).

В работе приводится описание нового дескриптора второго типа CPAFD (Color Periphery Area Fields Descriptor), учитывающего не только цветовую гамму изображения, но и геометрическую структуру изображения по всем цветам, то есть его фрактальность. Проводится оценка эффективности работы дескриптора с известными из литературы [16].

## 2. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

Будем считать, что классифицируемые изображения образуют множество  $X$ . Множество  $X$  называется множеством объектов. Множество классов, к которым может быть отнесен объект из  $X$ , обозначим  $Y$ . Множество  $Y$  называется множеством допустимых ответов.

Предполагается, что существует целевая функция (target function)  $y^t : X \rightarrow Y$ , значения которой  $y_i = y^t(x_i)$  известны на конечном множестве объектов  $x_1, \dots, x_l \subset X$ . Пары «объект — ответ»  $(x_i, y_i)$  называются прецедентами. Совокупность пар  $X^l = (x_i, y_i)_{i=1}^l$  называется обучающей выборкой.

Задача классификации по прецедентам состоит в следующем. Требуется по выборке  $X^l$  восстановить зависимость  $y^t$ , то есть построить решающую функцию (decision function)  $a; X \rightarrow Y$ , которая приближает целевую функцию  $y^t(x)$  не только на объектах обучающей выборки, но и на всем множестве  $X$ .

Признак (feature)  $f$  объекта  $x$  есть результат измерения некоторой характеристики объекта. Формально признаком называется отображение  $f : X \rightarrow D_f$ , где  $D_f$  — множество допустимых значений признака.

Набор признаков образует вектор  $F = (f_1, \dots, f_n)$ , который называют признаковым описанием объекта  $x \in X$ .

## 3. ПОСТАНОВКА ЗАДАЧИ

Дано множество объектов  $X$  и множество допустимых ответов  $Y$ . Существует целевая функция (target function)  $y^t : X \rightarrow Y$ , значения которой  $y_i = y^t(x_i)$  известны только на конечном множестве объектов  $x_1, \dots, x_l \subset X$ .

Требуется построить решающую функцию  $a : X \rightarrow Y$ , которая аппроксимирует целевую функцию  $y^t(x)$  на всем множестве  $X$ .

Процесс классификации принято разделять на три этапа: предобработку данных, сегментацию данных и их классификацию.

## 4. ПРЕДОБРАБОТКА ИЗОБРАЖЕНИЯ

В исходном изображении, как правило, содержится большое количество информации, которая является избыточной для решения задачи классификации. Основной задачей этапа предобработки является отсечение этой информации. Следует заметить, что за счет уменьшения объемов обрабатываемых данных уменьшается время классификации.

На этапе предобработки изображений количество избыточной информации уменьшается при переходе к чистым цветам. Заметим, что при этом сохраняются характеристики, учитывающие структурную неоднородность изображений.

Программа-фильтр, осуществляющая переход к чистым цветам, каждому пикселю входного изображения ставит в соответствие новый пиксель одного из восьми чистых цветов, в зависимости от яркости rgb-компонент входного изображения.

Фильтр работает по следующему принципу: если rgb-компонента имеет яркость больше 127, то ее значение становится 255; если меньше 127, то 0.

Результаты применения фильтра к изображению яхты показаны на рис. 1.



Рис. 1. Изображение яхты до и после фильтрации

На следующем этапе предобработки выполняется сбор информации об изображении:

- Сегментация изображения и определение областей связности для каждого цвета.
- Средняя площадь областей связности для каждого цвета.
- Средний периметр областей связности для каждого цвета.

## 5. СЕГМЕНТАЦИЯ И ВЫДЕЛЕНИЕ ГЕОМЕТРИЧЕСКОЙ СТРУКТУРЫ

Под сегментацией изображения понимают задачу разбиения плоскости образа на ряд связных непересекающихся областей, каждая из которых обладает внутренней однородностью того или иного вида (например цвет пикселей).

**Определение.** Пусть вся плоскость изображения обозначается как  $R$ . Тогда процедурой сегментации изображения называется процесс разбиения  $R$  на такую совокупность связных областей  $R_i, i = \overline{1, n}$  для которых выполняются следующие условия:

$$а) R = \bigcup_{i=1}^n R_i,$$

$$б) R_i \cap R_j = \emptyset, \forall i \neq j,$$

в)  $\text{Pred}(R_i) = \text{true}, i = \overline{1, n}$ , где  $\text{Pred}(R_i)$  — булевский предикат однородности области в некотором заданном смысле. В нашем случае  $\text{Pred}(R_i)$  принимает истинное значение, когда область связности  $R_i$  окрашена в один из 8 чистых цветов.

Первое условие данного выражения утверждает, что области разбиения целиком покрывают кадр изображения. Второе условие описывает тот факт, что области разбиения попарно не пересекаются. Третье условие требует, чтобы каждая область разбиения была однородной областью изображения в заданном смысле (например в смысле максимально допустимой разницы значений яркости составляющих ее пикселей).

В основе определения областей однородности и их границ лежит классический двух-проходной алгоритм поиска областей связности [3, 12] с последующим отбросом внутренних пикселей. Иллюстрация алгоритма и его описание приведены ниже.

## 6. ОПИСАНИЕ АЛГОРИТМА ОПРЕДЕЛЯЮЩЕГО ОБЛАСТИ СВЯЗНОСТИ

Возьмем некоторое изображение, прошедшее предобработку (рис. 2). На нем будут присутствовать только чистые цвета.

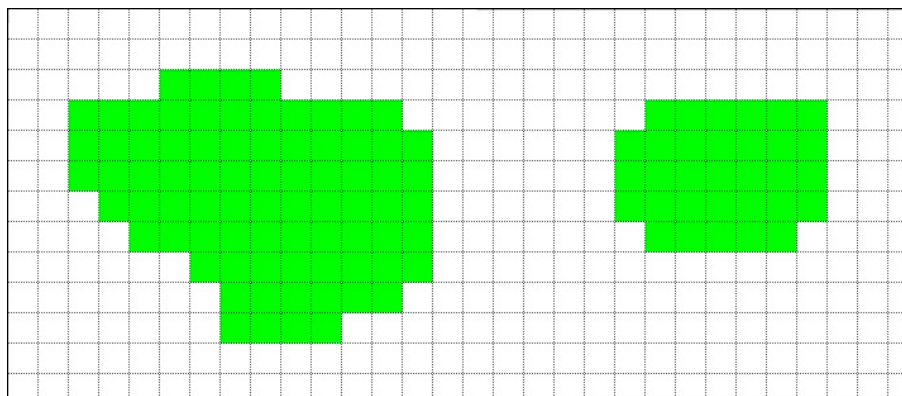


Рис. 2. Исходное изображение, прошедшее предобработку

### Первый проход алгоритма

Будем искать области связности зеленого цвета. Все остальные цвета будем считать цветом фона (в данном случае только белый). Начнем построчный обход изображения из левого верхнего угла слева направо. Если встречается пиксель зеленого цвета, то ему присваиваем индекс области, к которой он предположительно относится. Индексы начинаются с единицы. Если следующий пиксель также окрашен в зеленый цвет, то ему присваивается тот же индекс. Продолжаем проход по строке до тех пор, пока не встретится пиксель фоновой цвета. Пиксели фоновой цвета пропускаем. Если после фоновой пикселя вновь встречается пиксель зеленого цвета, то присваиваемый ранее индекс увеличиваем на единицу. Этот проход продолжаем до конца строки. Далее переходим на новую строку. Повторяем выше описанное до тех пор, пока не пройдем все изображение. В результате получим размеченное изображение (рис. 3).

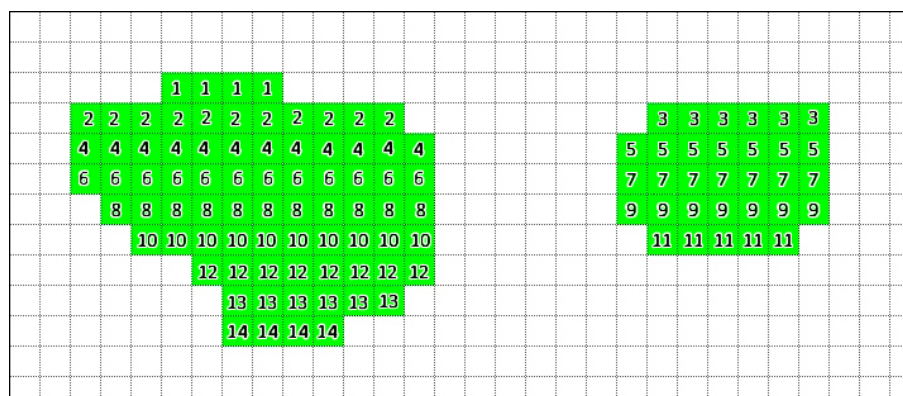


Рис. 3. Размеченное изображение

### Второй проход

Каждый пиксель изображения получил свою метку, для того чтобы выделить область связности одного цвета, следует устранить коллизии — ситуации, при которых

на одной области связности находятся различные индексы областей. Это означает, что следует снова пройти построчно все изображение и присвоить каждому пикселю индекс, минимальный среди соседей. Повторяем эту процедуру до тех пор, пока за полный проход изображения не произойдет присвоения новых индексов пикселям. Полученный результат показан на рис. 4.

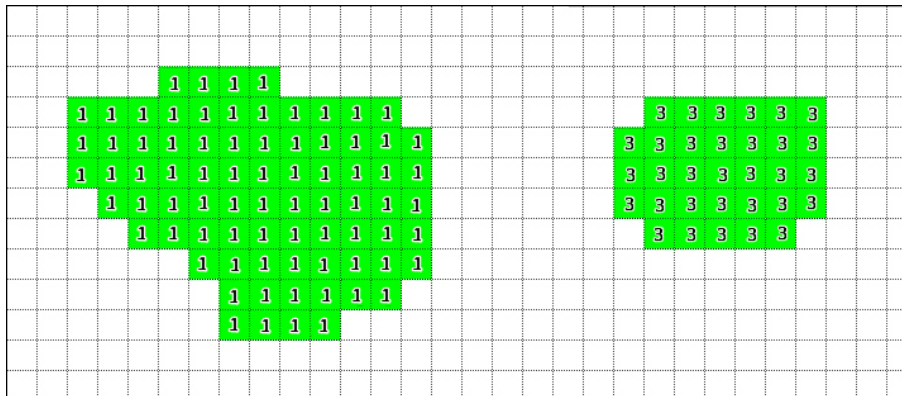


Рис. 4. Размеченное изображение, на котором устранены все коллизии

После устранения коллизий ищем границы областей связности. Для этого удаляем все внутренние пиксели зафиксированного цвета и перенумеровываем области таким образом, чтобы самый старший индекс обозначал не только последнюю найденную область связности, но и общее количество найденных областей (рис. 5). К внутренним пикселям относятся те пиксели, которые имеют соседей того же цвета по всем 4 направлениям (вверх, вниз, вправо, влево).

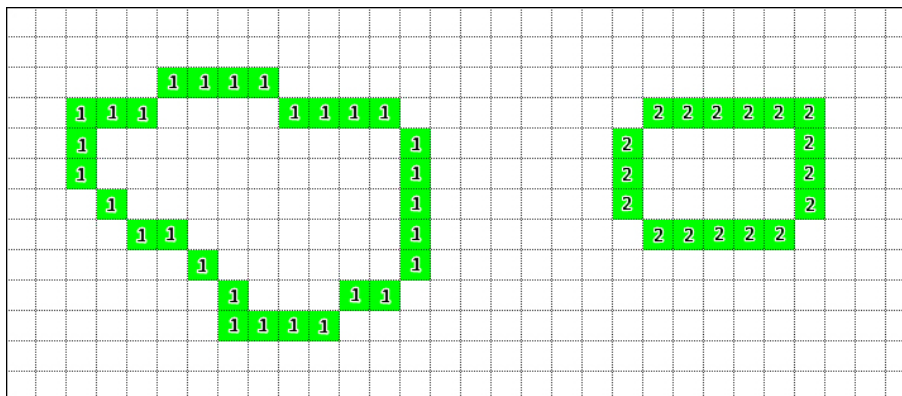


Рис. 5. Границы областей связности

В результате работы алгоритма получаем объекты (в терминах объектно-ориентированного программирования) границ областей связности для конкретного чистого цвета, в которых содержатся данные о границе: номер области связности, пиксели, образующие границу области. Используя эти значения, можно определить компоненты вектора признаков. Количество областей связности равно числу объектов границ; средний периметр равен количеству всех хранящихся в объектах пикселей, деленному на количество областей; средняя площадь находится подсчетом всех пикселей конкретного цвета на изображении и последующим делением на количество областей.



Используя описанный алгоритм для всех остальных цветов, получим соответствующие компоненты признакового описания изображения.

## 7. ПОСТРОЕНИЕ КЛАССИФИКАТОРА ИЗОБРАЖЕНИЙ

Категоризация изображений осуществляется с помощью метрического классификатора, то есть алгоритма, осуществляющего классификацию поданного на вход программе объекта на основе его «близости» к прецедентам. Близость определяется следующей функцией расстояния:

$$p: X \times X \rightarrow [0, \infty).$$

В работе используется метрика «городских кварталов». Для этой метрики влияние отдельных больших разностей между компонентами признакового описания меньше, чем для метрик более высокого порядка.

$$p(u, x) = \sum_{j=1}^n |u^j - x_j^j|,$$

где  $u = (u^1, \dots, u^n)$ ,  $x_j = (x_j^1, \dots, x_j^n)$  — признаковые описания объектов,  $u$  — классифицируемый объект,  $x_j$  — объект обучающей выборки.

Идея возможности классификации на основе расстояния между объектами основывается на гипотезе компактности, в которой утверждается, что схожие объекты лежат в одном классе [10].

Классифицируемый объект подается на вход классификатору, который определяет расстояние между ним и всеми прецедентами. Найденные расстояния сортируются по возрастанию.

На основе гипотезы компактности можно сделать вывод о том, что, чем меньше найденное расстояние между объектом и  $i$ -ым прецедентом, тем больше шансов, что объект принадлежит к классу  $y_j$   $i$ -того прецедента. Чтобы формализовать это утверждение, вводят неотрицательную невозрастающую функцию веса  $K: [0; 1] \rightarrow [0; 1]$ . В работе используется квадратичное ядро  $K(r) = (1-r^2)^2$ , где  $|r| < 1$ . Для уменьшения вероятности ложной классификации будем учитывать расстояние и вес не только до ближайшего «соседа», а до  $k$  соседей. Классификатор приобретает следующий вид:

$$\sum_{i=1}^k [y^i = y] \cdot K\left(\frac{p(u, x^i)}{p(u, x^{k+1})}\right),$$

Классифицируемый объект будет отнесен к тому классу, на котором он имеет наибольший вес среди  $k$  ближайших соседей в окрестности радиуса  $p(u, x^{k+1})$ .

Классифицирование на основе данного алгоритма называется методом окна Парзена переменной ширины.

Качество работы данного метода зависит от количества  $k$  соседей, с которыми сравнивается объект  $u$ . Чтобы качество классификации было максимальным, требуется обучить классификатор, то есть найти оптимальное  $k$ , при котором количество ошибок будет минимально.

## 8. ОБУЧЕНИЕ КЛАССИФИКАТОРА И ОТБОР ЭТАЛОННЫХ ИЗОБРАЖЕНИЙ

*Метод обучения* (learning algorithm) — это отображение:  $(X \times Y)^l \rightarrow A$ , которое ставит в соответствие произвольной выборке  $X^l = (x_i, y_i)_{i=1}^l$  некоторый алгоритм  $a \in A$ . Говорят, что метод строит алгоритм  $a$  по выборке  $X^l$  [8].

*Функция потерь* (loss function) — это неотрицательная функция  $a$ , характеризующая величину ошибки алгоритма  $a$  на объекте  $X$ . Если  $L(a, x) = 0$ , то ответ  $a(x)$  называется корректным.

Функционал качества алгоритма  $a$  на выборке  $X^l$ :  $Q(a, X^l) = \frac{1}{l} \sum_{i=1}^l L(a, x_i)$ .

Пусть дана выборка  $X^l = (x_i, y_i)_{i=1}^l$ . Разобьём её  $N$  различными способами на две непересекающиеся подвыборки: обучающую  $X_n^l$  длины  $l$  и контрольную  $X_n^k$  длины  $k = L - l$ . Для каждого разбиения  $n = 1, \dots, N$  построим алгоритм  $a_n = \mu(X_n^l)$  и вычислим значение  $Q(a_n, X_n^k)$ . Среднее арифметическое значений  $Q_n$  по всем разбиениям называется оценкой скользящего контроля (cross-validation, CV):

$$CV(u, X^l) = \frac{1}{N} \sum_{n=1}^N Q(\mu(X_n^l), X_n^k).$$

В работе реализована процедура поиска оптимального параметра  $k$  ближайших соседей, при котором кросс-валидация стремится к минимуму. Это значит, что ищется такое  $k$ , при котором количество ошибок при скользящем контроле минимально.

$$\frac{1}{N} \sum_{n=1}^N Q(a_k(X^l), X_n^M),$$

где  $1 < k < L - 1$ ,  $a_k$  — классификатор с параметром  $k$  ближайших соседей,  $X^l$  — обучающая подвыборка,  $X_n^M$  — контрольная подвыборка.

Объекты обучающей выборки принято разделять на четыре класса: эталонные, неинформативные, периферийные и шумовые.

*Эталонные объекты* — это объекты, которые в пространстве признаков находятся близко друг к другу и далеко от разделяющей поверхности. Понятие удаленности объектов зависит от выбранной метрики и признаков, которые должны удовлетворять гипотезе компактности.

Отбор эталонов основан на введении функции отступа (margin), характеризующей степень типичности объекта обучающей выборки для своего класса. Величина отступа для объекта  $x_i \in X^l$  относительно классификатора вида  $a(u) = \arg \max_{y \in Y} \Gamma_y(u)$ , определяет-

ся как  $M(x_i) = \Gamma_{y_i}(x_i) - \max_{y \in Y, y_i} \Gamma_y(x_i)$ , где  $\Gamma_y(u) = \sum_{i=1}^k [y^j = y] \cdot K\left(\frac{p(u, x^i)}{p(u, x^{k+1})}\right)$  — суммарный вес объекта  $u$  в классе  $y \in Y$ .

*Неинформативные объекты* имеют положительный отступ, но практически не добавляют новой информации о своем классе по сравнению с эталонными объектами.

*Периферийные объекты* находятся близко к разделяющей поверхности. Классификация периферийных объектов неустойчива в том смысле, что при малом изменении параметров классификатора или обучающей выборки она может меняться. Отбрасывание периферийных объектов практически не влияет на качество классификации.

*Шумовые объекты (выбросы)* — это объекты, которые, несмотря на принадлежность к тому же классу, что и эталонные, классифицируются неверно. Например, при составлении каталога с изображениями яхт в него попали изображения с животными. В этом

случае изображения с животными будут относиться к классу яхт, но иметь отличные от яхт векторы признаков описаний, что снизит качество классификации яхт.

Разделение типов объектов по их отступам показано на рисунке 6.

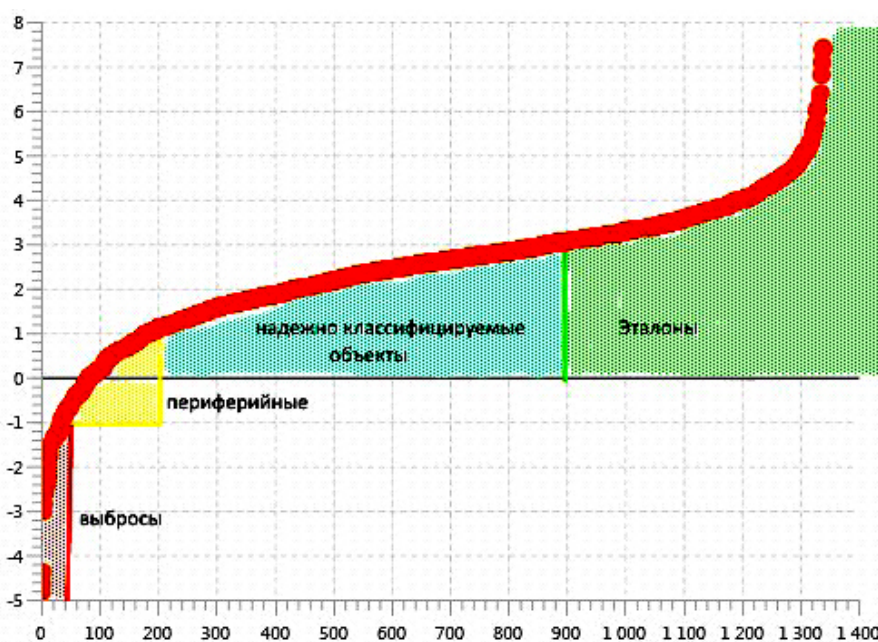


Рис. 6. Разделение типов объектов по их отступам

## 9. ТЕСТИРОВАНИЕ СИСТЕМЫ КЛАССИФИКАЦИИ НА ОСНОВЕ CRAFT

Для проведения тестирования качества классификации на основе CRAFT дескриптора была разработана система Eyesimo. Исходные коды Eyesimo можно свободно скачать с GitHub.

### Тестирование алгоритма сегментации

Тестирование системы Eyesimo проводилось на изображениях небольшого размера, содержащих области разной степени сложности (рис. 7). Результаты одного из тестов показаны ниже.

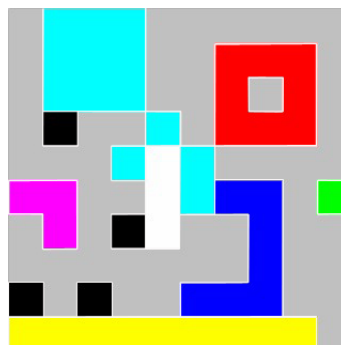


Рис. 7



Векторы признаков описаний изображений, записанные в удобной для восприятия форме имеют вид:

Size: 10 × 10			
	fields	midArea	midPeriphery
red	1	8	8
green	1	1	1
blue	1	7	7
magenta	1	3	3
yellow	1	9	9
cyan	4	3	3
black	4	1	1
white	1	3	3

Здесь Size  $W \times H$  — ширина и высота изображения (в пикселях), fields — количество областей связности, midArea — средняя площадь (в пикселях), midPeriphery — средний периметр (в пикселях). Корректность результата можно проверить непосредственным подсчетом.

Ниже показан вектор признакового описания изображения яхты на рис. 1, записанный в удобной для восприятия форме:

Size: 326 × 215			
	fields	midArea	midPeriphery
red	0	0	0
green	1	1	1
blue	482	66	15
magenta	7	1	1
yellow	0	0	0
cyan	843	20	5
black	377	10	8
white	283	61	16

## 10. АНАЛИЗ КАЧЕСТВА КЛАССИФИКАЦИИ СИСТЕМЫ EYESIMO

Для проведения тестирования была создана база из 10000 изображений, на которых присутствуют объекты 7 разных классов: автомобили, кошки, собаки, еда, люди, самолеты, яхты. На каждый класс приходится примерно равное количество изображений одинакового размера.

Методология тестирования:

- Качество классификации оценивается как процент успешности, который равен и округляется до целого числа. CV — кросс-валидация.
- Классификация ведется по двум классам: основному, в котором присутствуют только интересующие нас объекты, и вспомогательному, в котором находятся любые объекты, кроме интересующих.
- Количество объектов в каждом классе обучающей выборке примерно одинаково.
- Общее количество объектов обучающей выборки до отбора эталонов около 3000.
- Все изображения из обучающей выборки приводятся к размеру 300 пикселей по ширине и 200 по высоте.

- Требование стандартного размера необходимо для корректного сбора признаков и поиска метрик между объектами.
- Выбранный размер связан с размером исходных изображений — у всех он примерно равен 300x200.
- Тестируется классификация 7 разных типов объектов: auto (автомобили), cats (кошки), dogs (собаки), food (еда), people (люди), planes (самолеты), yachts (яхты). Для каждого имеется  $\approx 1500$  изображений. В основном классе находится 1500 изображений с интересующим типом объектов; в дополнительном классе находится  $\approx 250$  изображений каждого из 6 оставшихся классов.
- Для того чтобы избавиться от шумовых и пограничных объектов и при этом снизить риск появления эффекта переобучения системы, отбор происходит с отступом 0,5.
- Измерение кросс-валидации берется для обученного параметра  $k$ .

Результаты тестирования (рис. 8) показывают, что средняя успешность классификации до отбора эталонов составляет 77 %; после отбора эталонов с порогом в 0.5 успешность составляет 97 %. Это значит, что в среднем 23 % новых объектов, не входящих в обучающую выборку, будут классифицированы неверно. Если на вход классификатору подается типичный представитель собственного класса, то средняя ошибка классификации 3 %.

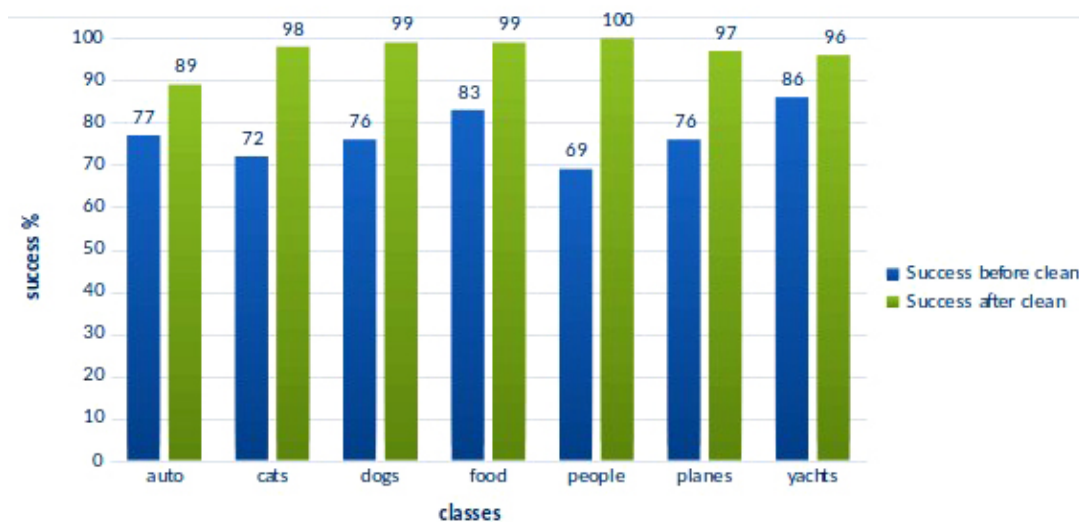


Рис. 8. График успешности классификации изображений до и после отбора эталонов

В работе [16] проводилась классификация 1300 изображений некоторой поверхности с дефектами по 14 классам с использованием стандартных дескрипторов методом ближайших соседей.

При сравнении результатов классификации, полученных с помощью Eyesimo, с результатами из работы [16] нужно учитывать, что в этой работе не проводился отбор эталонов через функцию отступа. Следовательно, корректным будет сравнение результатов без учета отбора эталонов.

Если сравнивать результаты классификации по одному дескриптору, то SRAFD демонстрирует относительно высокие показатели качества классификации (77 %).

Если взять среднюю успешность классификации по всем дескрипторам (черные столбцы второго графика), то она 80 %.

## 11. РАСПОЗНАВАНИЕ ОБЪЕКТОВ В ВИДЕОПОТОКЕ

СРАFD дескриптор может быть использован для анализа видеопотока в реальном времени.

Eyesimo поддерживает анализ видеопотока. Во вкладке «обработка видео» пользователь должен выбрать видеофайл для анализа и нажать кнопку «начать». После запуска выполняется инициализация плеера, начинается захват кадров видео и их анализ. Новый кадр захватывается после окончания анализа предыдущего кадра. Результаты анализа записываются в соответствующем поле «лог обработки».

Показатель количества захватываемых и классифицируемых кадров в секунду зависит от стандартного размера изображения, к которому приводились изображения, из характеристик которых составлялась база характеристик конкретного класса объектов. Чем больше стандартный размер, тем больше времени требуется для обработки такого изображения алгоритмами предобработки и сегментации. Кроме того, скорость обработки зависит от мощности центрального процессора, поэтому показатели количества захватываемых и классифицируемых кадров в секунду будут различаться в каждой конфигурации Eyesimo. Информация о количестве обрабатываемых кадров в секунду отображается при запуске видео в поле FPS.

Пример анализа видео показан на рис. 9.

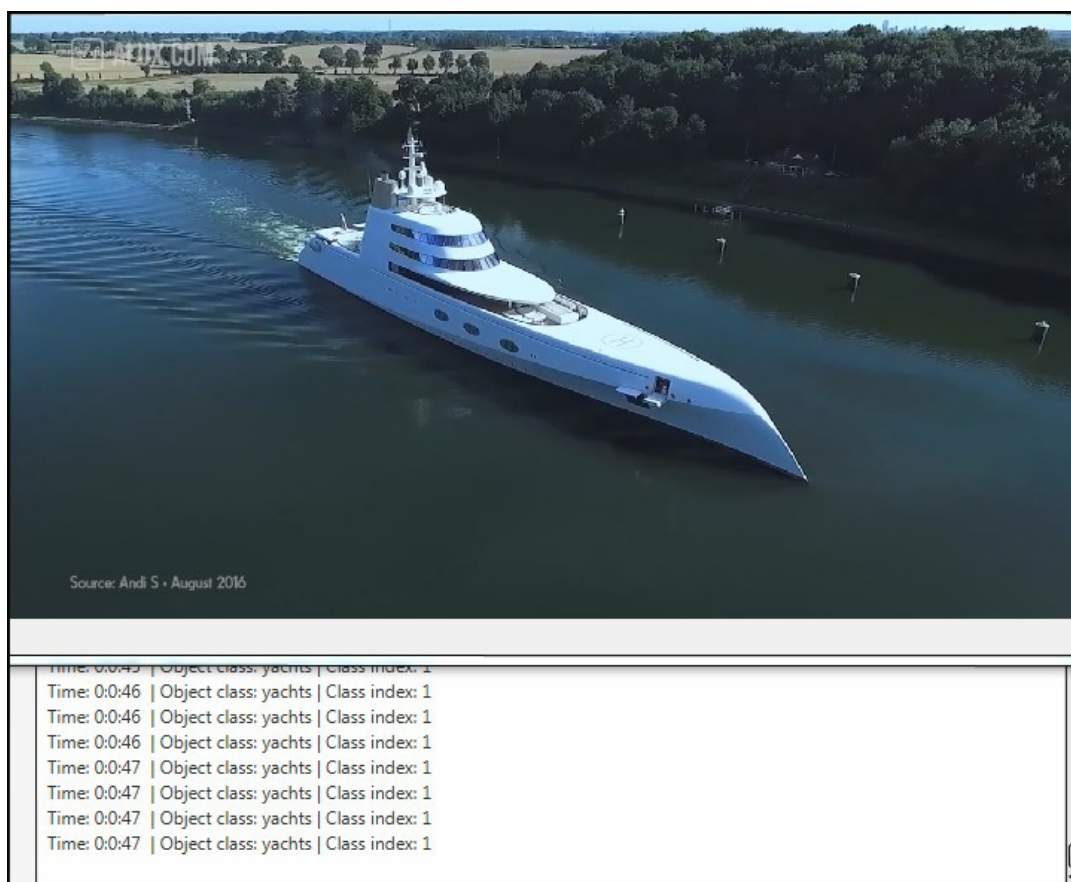


Рис. 9. Успешное обнаружение яхты

## 12. ЗАКЛЮЧЕНИЕ

Предложенный CRAFT дескриптор успешно прошел тестирование. С его помощью можно надежно классифицировать типичных представителей какого-либо класса. Качество классификации близко к результатам, которые показывают дескриптор.

Заметим, что созданный CRAFT дескриптор первого типа обладает гибкой структурой. Это позволяет изменять количество цветов, уменьшать и увеличивать размерность признакового описания для анализируемых классов объектов.

### Список литературы

1. *Потапов А. А.* Новейшие методы обработки изображений М.: Физматлит, 2008.
2. *Лепский А. Е., Бронвич А. Г.* Математические методы распознавания образов. Таганрог: Изд. ТТИЮФУ, 2009.
3. *Бабаян П. В., Фельдман А. Б.* Распознавание объектов на изображениях при наблюдении из космоса // Вестник РГТУГ. 2008. № 4 (26). С. 124–132.
4. *Вапник В. Н., Червонекис А. Я.* Теория распознавания образов (статистические проблемы обучения) М.: Наука, 1974.
5. *Козлов В. Н.* Элементы математической теории зрительного восприятия. М.: Изд. Центр прикладных исследований при мех-мате МГУ, 2001.
6. *Яне Б.* Цифровая обработка изображений. М.: Техносфера, 2007.
7. *Ляхов А. Ф.* Вероятностные методы оценки параметров изображения по существованию регулярных структур из пикселей // Компьютерные инструменты в образовании. 2009. № 5. С. 17–26.
8. *Петухов А. А.* Распознавание текстурных изображений на основе статических и фрактальных признаков // Вестник Сибирского ГАУ. 2011. С. 63–68.
9. *Каптанов Н. В., Ляхов А. Ф.* Фрактальная размерность визуального образа математической матрицы // Компьютерные инструменты в образовании. 2013. № 2. С. 59–66.
10. *Воронцов К. В.* Математические методы обучения по прецедентам. М: МФТИ, 2006.
11. *Захаров Р.* Метод классификации объектов различных классов на видео потоке и на статичных изображениях. URL: <http://docplayer.ru/54279516-Metod-klassifikacii-obektov-razlichnyh-klassov-na-video-potoke-i-na-statichnyh-izobrazheniyah.html> (дата обращения: 14.11.18).
12. *Rosenfeld A., Pfaltz P.* Sequential Operations in Digital Picture Processing // Journal of the Association for Computing Machinery. 1966. Vol. 13. P. 471–494.
13. *Deselaers T., Keysers D., Ney H.* Features for Image Retrieval: An Experimental Comparison / German Research Center for Artificial Intelligence (DFKI). Kaiserslautern, Germany. November 29, 2007
14. *Pakkanen J., Iivesmäki A., Iivarinen J.* Defect image classification with MPEG-7 descriptors // Proceedings of the 13th Scandinavian Conference on Image Analysis. Halmstad, Sweden, 2003. P. 349–355. (Lecture Notes in Computer Science, vol. 2749).
15. *Buturovic A.* MPEG 7 Color Structure Descriptor for visual information retrieval project VizIR // Institute for Software Technology and Interactive Systems Technical University Vienna, 2005. URL: <http://citeseerx.ist.psu.edu/viewdoc/versions?doi=10.1.1.369.8132> (дата обращения: 14.11.18).
16. *Rublee E., Rabaud V., Konolige K., Bradski G.* ORB: An efficient alternative to SIFT or SURF // IEEE International Conference on Computer Vision. November 2011 P. 2564–2571.

Поступила в редакцию 10.10.2017, окончательный вариант — 14.11.2017.

Computer tools in education, 2017

№ 6: 31–43

<http://ipo.spb.ru/journal>

## DESCRIPTION OF A DESCRIPTOR FOR THE CATEGORIZATION OF IMAGES TAKING INTO ACCOUNT THE GEOMETRIC STRUCTURES IN ALL COLORS

Lyakhov A. F.<sup>1</sup>, Chnegov E. I.<sup>1</sup>

<sup>1</sup>Lobachevsky University, Nizhny Novgorod, Russia

### Abstract

A new original descriptor is proposed to allow the categorization, of both static images and video streams. The descriptor takes into account the color gamut of the image and the geometric structure for all colors, that is, the fractalness of the image. A comparison is presented of the efficiency of this descriptor with the description of the descriptors in the literature.

**Keywords:** *descriptor, pattern recognition, image classification.*

**Citation:** A. F. Lyakhov and E. I. Chnegov, "Description of a Descriptor for the Categorization of Images Taking into Account the Geometric Structures in All Colors," *Computer tools in education*, no 6, pp. 31–43, 2017 (in Russian).

*Received 10.10.2017, the final version — 14.11.2017.*

**Alexander F. Lyakhov, Associate Professor, Lobachevsky University, Institute of Information Technologies, Mathematics and Mechanics Department of Theoretical, Computer, Experimental Mechanics; 603062 N.Nizhniy Novgorod, Autumn passage, 106 [Lyakhov@mm.unn.ru](mailto:Lyakhov@mm.unn.ru)**

**Yegor I. Chnegov, undergraduate, Lobachevsky University, Institute of Information Technologies, Mathematics and Mechanics Department of Theoretical, Computer, Experimental Mechanics, [kizlarov@yandex.ru](mailto:kizlarov@yandex.ru)**

---

**Ляхов Александр Фёдорович,**  
кандидат физико-математических наук,  
доцент, ННГУ, Институт Информационных  
технологий, математики и механики  
Кафедра теоретической, компьютерной,  
экспериментальной механики; 603062  
г. Нижний Новгород, Осенний проезд, 10,  
[Lyakhov@mm.unn.ru](mailto:Lyakhov@mm.unn.ru)

**Чнегов Егор Игорьевич,**  
магистрант второго года обучения, ННГУ,  
Институт Информационных технологий,  
математики и механики Кафедра  
теоретической, компьютерной,  
экспериментальной механики,  
[kizlarov@yandex.ru](mailto:kizlarov@yandex.ru)

©

Наши авторы, 2017.

Our authors, 2017.