

Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Казанский (Приволжский) федеральный университет»

**ИНСТИТУТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

КАФЕДРА СИСТЕМНОГО АНАЛИЗА И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Направление: 010400.62 Прикладная математика и информатика
Профиль: Системное программирование

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(Бакалаврская работа)**

**Разработка и реализация метода классификации изображений по
представленным сценам**

Работа завершена:

«___» _____ 2015 г. _____ Б.А. Насрулин

Работа допущена к защите:

Научный руководитель:

К.ф.-м.н., ассистент кафедры системного анализа и информационных технологий КФУ

«___» _____ 2015 г. _____ Е.В. Разинков

Заведующий кафедрой

Д.т.н., профессор

«___» _____ 2015 г. _____ Р.Х. Латыпов

Казань – 2015 год

ВВЕДЕНИЕ	3
1. МЕТОДЫ КЛАССИФИКАЦИИ СЦЕН.....	5
1.1. Задача классификации изображений	5
1.2. Понятие сцены и ее классификации	7
1.3. Подходы в классификации сцен	8
2. МЕТОДЫ БЫСТРОЙ КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ	10
2.1. Быстрый классификатор.....	10
2.1.1. Алгоритм <i>AdaBoost</i>	10
2.1.2. Базовые классификаторы в <i>AdaBoost</i>	12
2.2. Быстрый дескриптор изображения	13
2.2.1 <i>ORB</i>	13
2.2.2 <i>HOG</i>	14
3. ПРОГРАММНАЯ СИСТЕМА И ЭКСПЕРИМЕНТЫ	17
3.1. Программная система	17
3.2. Эксперименты с базой изображений	19
3.2.1 <i>База изображений</i>	19
3.2.2 <i>Проведенные эксперименты</i>	20
3.3. Эксперименты в реальном времени	23
ЗАКЛЮЧЕНИЕ	25
СПИСОК ЛИТЕРАТУРЫ.....	26

ВВЕДЕНИЕ

В настоящее время идет бурное развитие вычислительных систем и информационных технологий как в науке и индустрии, так и в повседневной жизни. Это привело к резкому увеличению потоков данных и необходимости их эффективной обработки.

Одним из наиболее популярных направлений обработки данных является компьютерное зрение и интеллектуальная обработка изображений, а именно большое внимание уделяется классификации изображений по представленным сценам. Распознавание сцены изображения применяется во многих сферах компьютерного зрения: калибровка камеры, распознавание объектов в контексте местоположения, построение визуальной карты робота.

Несмотря на то, что в решении задачи классификации объектов на изображениях были достигнуты значительные успехи, задача классификации сцен на изображениях решена не полностью, т.к. для работы алгоритма необходимы большие компьютерные мощности и распараллеливание на графических процессорах.

Целью данной работы является создание универсального алгоритма, который не зависит от наличия графического процессора и осуществляет быструю классификацию. Так, например, будет возможна работа на слабых компьютерах и мобильных устройствах в режиме реального времени.

Для достижения данной цели необходимо выполнить ряд задач:

- Проанализировать существующие подходы для решения проблемы классификации изображения по представленным сценам;
- Выбрать оптимальный подход для создания системы по классификации изображений;
- Реализовать систему, которая классифицирует изображения;
- Протестировать разработанную систему;
- Сделать вывод, основываясь на результатах.

В первом разделе работы описана постановка задачи, а также обзор существующих решений.

Во втором разделе рассматривается наш подход к решению проблемы распознавания сцены.

В третьем разделе описывается реализованная программная система и приводятся проведенные эксперименты.

1. МЕТОДЫ КЛАССИФИКАЦИИ СЦЕН

1.1. Задача классификации изображений

Распознавание сцен по изображениям относится к известному разделу машинного обучения, который называется задачей классификации.

Задача классификации

Данная задача решается при заданном множестве объектов, разделенных на конечное количество классов. Это множество называется *обучающей выборкой*.

Выборка, не содержащая объектов из обучающего множества, которую мы будем использовать во время тестирования и оценки работы алгоритма классификации, называется *тестовой*.

Требуется построить алгоритм, способный классифицировать произвольный объект из тестовой выборки.

По каждому из объектов строится набор признаков, который представляет собой числовой вектор фиксированной длины n . Процесс *обучения* алгоритма сводится к применению численных методов оптимизации и нахождению оптимальных параметров, разделяющих множества на классы.

Объект можно классифицировать по его вектору признаков.

Для вычисления вектора признаков изображения используются *детекторы* и *дескрипторы* изображения [1].

Операция сравнения изображений по всем точкам занимает множество времени. Для оптимизации этого процесса достаточно сравнить ключевые точки изображений. Ключевая точка изображения – это область изображения, выделяющаяся среди остальных наличием определенных признаков (резкий перепад освещенности, другой цвет, другой угол). Для обнаружения и выявления этих ключевых точек используются детекторы.

Детектор вычисляет позицию ключевых точек, причем вычисление позиций должно быть инвариантно относительно поворота и масштаба изображения.

Получив набор ключевых точек для изображений, можно начать сравнивать их. Для сравнения изображений по их ключевым точкам используется дескриптор, который описывает особенность ключевой точки с помощью вектора значений. Полученный вектор значений представляет собой абсолютную величину, с помощью которой можно распределить, упорядочить изображения относительно какого-либо признака.

После получения описания ключевых точек необходимо нормализовать все вектора значений дескрипторов и привести их к вектору признаков фиксированной длины.

Теперь вместо того, чтобы на обучение подавать всё изображение, будет использоваться вектор признаков, вычисленный на данном изображении. Причем каждому изображению, независимо от его размеров, соответствует вектор одной и той же длины.

Для оценки работы алгоритма вводится специальный параметр, который является индикатором разделения данных на заданные классы. В роли такого индикатора в нашей работе используется коэффициент точности, который представляет собой отношение правильно классифицированных объектов к общему числу объектов на тестовом множестве.

1.2. Понятие сцены и ее классификации

Прежде всего нам необходимо определить, что мы понимаем под сценой, представленной на изображениях. Четкое определение сцены было представлено в работе [2]. По мнению авторов, сцену легче всего определить в контексте поиска объектов. Как правило, поиск объекта осуществляется на расстоянии 1-2 метра от “обозревателя”, но при классификации сцены важными становятся также и объекты, находящиеся на расстоянии более пяти метров, иными словами, общий глобальный контекст. Таким образом, сцена - это глобальное представление пространства, в котором находится наблюдаемый объект.

Для классификации сцены необходимо выбрать способ ее описания. Выделяют два подхода к описанию сцены: “сверху-вниз” и “снизу-вверх”. В рамках подхода “снизу-вверх” описание сцены строится с помощью объектов и мелких деталей на изображении.

В итоге, сцену можно представить, как совокупность объектов. В подходе “сверху-вниз” напротив, используются глобальные характеристики изображения.

Использование подхода “снизу - вверх” может быть более сложной задачей в терминах скорости вычисления, так как требуется поиск и распознавание объектов. Поэтому в нашей работе используется подход “сверху вниз”, в котором мы будем работать с характеристиками самого изображения, а не с совокупностью объектов.

Еще одной нетривиальной задачей является выделение классов сцен. В качестве классов для разделения сцен в работе [2] были предложены “значимые категории”. Значимые категории должны достаточно отличаться друг от друга, так чтобы можно было однозначно определить, к какому классу относится изображение. В качестве базовых значимых категорий можно выделить классы “внутри помещения” и “вне помещения”, которые были впервые предложены Тверски и Хеменвеем [3]. Также были предложены и

другими базовые значимые категории: искусственный (сделанный человеком) и натуральный [4]. Существуют подтверждения, что эти категории используются в человеческом зрении [4].

Определение базовых категорий изображения является первым шагом классификации. После этого можно начать классификацию на подкатегории, учитывая базовую, например, на город, парк, гору, пляж [2].

В данной работе в качестве базовых категорий рассматриваются классы “внутри помещения”, “вне помещения”.

1.3. Подходы в классификации сцен

Рассмотрим существующие подходы к классификации изображений по двум категориям: “внутри помещения” и “вне помещения”.

Одна из первых попыток классификации была предпринята в работе [5]. Их подход заключался в разделении изображения на блоки и оценке доминирующей ориентации линии на этих блоках. Доминирующая ориентация линий представляет собой угол поворота большинства линий в рамках одного блока. Набор ориентаций используется в качестве характеристики изображения.

Применение цвета и ориентации как характеристик было предложено в [6]. В качестве классификатора был выбран метод опорных векторов [9]. Однако, в данной работе не был предложен универсальный способ выбора характеристик сцены, точность сильно зависела от представленных изображений, их масштаба и ориентаций.

Одним из стандартных дескрипторов, описывающих глобальные характеристики изображения, стал GIST [7]. GIST основан на применении фильтров Габора [8] на различных масштабах. Данный подход в комбинации с классификатором позволяет добиться точных результатов в классификации. Главным недостатком этого метода является чрезмерная

длительность вычисления характеристик изображения, особенно для больших изображений.

Популярный в последнее время метод Глубинного обучения [8] показал очень точные результаты в классификации изображений по представленным сценам [9, 10]. Глубинное обучение представляет собой использование больших нейронных сетей с множеством скрытых слоев. Однако, значительным недостатком этого метода является очень длительное и сложное обучение модели с использованием больших кластеров и распараллеливание на графических процессорах.

2. МЕТОДЫ БЫСТРОЙ КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ

2.1. Быстрый классификатор

2.1.1. Алгоритм AdaBoost

Для того, чтобы добиться быстрой классификации изображений нужно ускорить вычисление детектора и дескриптора. Дескриптор для нашей

В качестве быстрого классификатора в данной работе был выбран алгоритм AdaBoost, который позволяет добиться точной классификации, быстро обучаться и тестировать в реальном времени.

Алгоритм AdaBoost — алгоритм машинного обучения, который был впервые предложен Йоавом Фройндом и Робертом Шапиром [10]. Алгоритм сводится к построению композиций из базовых алгоритмов обучения для улучшения эффективности классификации простых классификаторов. Каждый следующий классификатор строится по объектам, которые плохо классифицируются предыдущими.

AdaBoost основан на вызове слабых классификаторов в цикле. После каждого вызова распределение весов, отвечающих за важность того или иного объекта из обучающего множества, обновляется. Вес неверно классифицированного объекта возрастает, и новый классификатор ориентируется на объекты, в которых ошиблись предыдущие.

При данной обучающей выборке X^N строятся базовые алгоритмы классификации b_1, \dots, b_N .

Процесс вычисления можно разбить на несколько шагов:

1. Каждому объекту инициализируются начальные веса.
2. Находится классификатор $b_i: X \rightarrow \{-1, +1\}$, который минимизирует взвешенную ошибку классификации $b_i = \operatorname{argmin} Q(b, W^l)$

3. Коэффициент взвешенного голосования по классификации пересчитывается: $\alpha_t = \frac{1}{2} \ln \frac{1 - Q(b, W^t)}{Q(b, W^t)}$
4. Каждый вес пересчитывается. $w_i = w_i \exp(-\alpha_t y_i b_t(x_i)) \quad i = 1, \dots, l$
5. Веса объектов нормируются.
6. Алгоритм возвращает $a(x) = \text{sign}(\sum_{i=1}^N a_i b_i(x))$.

Достоинства алгоритма AdaBoost является хорошая обещающая способность, с помощью нее удается построить композиции, которые позволяют получить более точные результаты по классификации. Точность классификации должна увеличиваться по мере увеличения числа базовых алгоритмов. Время бустинга и работы AdaBoost практически сводится ко времени обучения базовых алгоритма, что позволяет получить очень быстрый вариант обучения классификатора. Так же стоит отметить, что алгоритм AdaBoost способен классифицировать так же шумы, которые возможны при классификации больших баз данных изображений.

Однако, алгоритм бустинга также имеет ряд недостатков. Так например, AdaBoost склонен к переобучению на базах данных, содержащих значительное число шума. Это связано с тем что, функция потерь слишком сильно увеличивает вес объектов, на которых ошибаются базовые классификаторы. На практике, такие объекты чаще всего оказываются выбросами. Алгоритм AdaBoost настраивается на шум, что приводит к переобучению. Данную проблему, можно решить предварительной обработкой данных, убрав эти шумы, или используя другие, более гибкие варианты базовых классификаторов, реагирующие на эти выбросы. AdaBoost требует достаточно большой базы для обучения. Также, для построения больших композиций, состоящих из значительного количества базовых классификаторов (более 100), требуется наличие большого количества оперативной памяти для хранения базовых классификаторов и их обучения.

2.1.2. Базовые классификаторы в AdaBoost

На вход алгоритму AdaBoost подаются слабые классификаторы, их количество.

В одной из ранних версий AdaBoost, слабые классификаторы возвращали лишь один из конечных классов, например $\{-1, +1\}$. В конечном итоге строилась композиция, которую также называют дискретным AdaBoost (Discrete AdaBoost).

Однако, на практике используется улучшенный вариант AdaBoost, называемый Вещественный AdaBoost (Real AdaBoost).

Алгоритм вещественного AdaBoost выдает вероятность $p(x) = P(y = 1|x)$, вероятность того что x принадлежит к определенному классу. В работе[11] была предложена схема вычисления вещественного AdaBoost.

В работе [11] была проведено сравнение разных классификаторов, дискретного и вещественного AdaBoost. Результаты представленные на рисунке 1, показывают, что наилучший результат классификации дает вещественный вариант алгоритма AdaBoost. В качестве слабых классификаторов лучшим вариантом оказались деревья решения с малой глубиной дерева (с глубиной 2, 3).

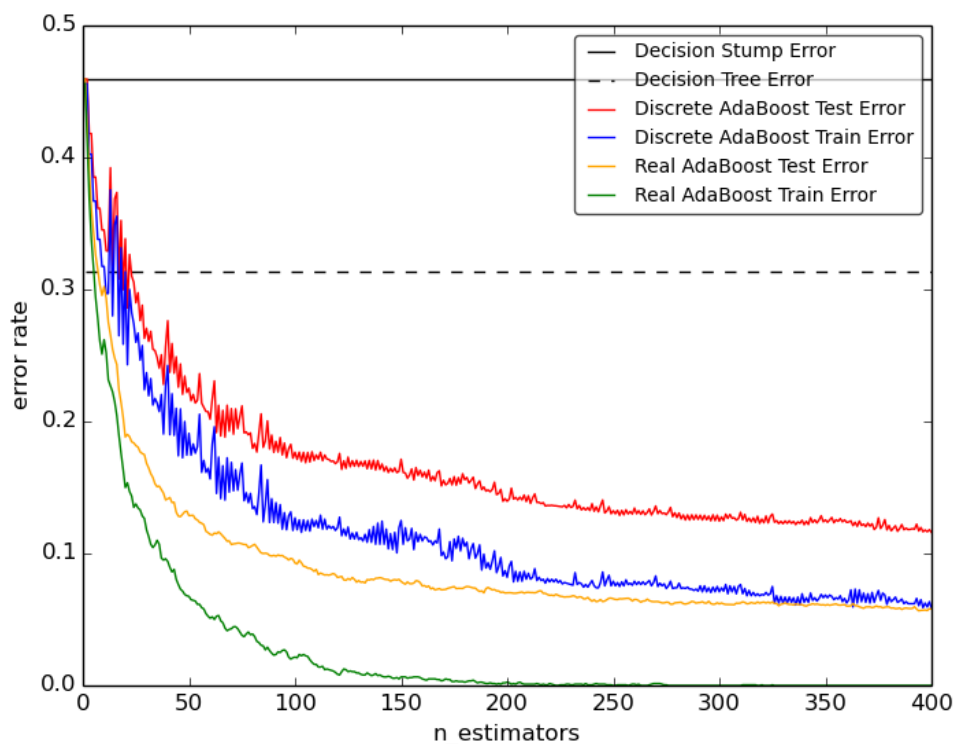


Рис. 1. Результаты сравнения различных вариантов AdaBoost и Деревьев решения.

2.2. Быстрый дескриптор изображения

2.2.1 ORB

ORB (Oriented FAST and Rotated BRIEF) – это быстрый робастный дескриптор и детектор, который часто используется для распознавания объектов. Он основан на визуальном дескрипторе BRIEF (Binary Robust Independent Elementary Features) [12] и детекторе FAST (Features from accelerated segment test) [14]. Целью создания данного дескриптора и детектора являлось построение эффективной замены SIFT [15].

Основной мотивацией авторов ORB являлось создание эффективного детектора и дескриптора, вычисляемых на стандартных CPU. ORB использует детектор oFAST и дескриптор rBRIEF.

FAST принимает на вход в качестве одного из параметров порог интенсивности между центральным пикселем и пикселями в круге определённого радиуса. Полученные точки сортируются по убыванию разности интенсивностей между рассматриваемым пикселем и центральным, согласно мере Харриса и выбираются первые N точек.

Данная процедура повторяется на разных масштабах. Таким образом, получаются характеристики FAST для разных масштабов.

Для того, чтобы добиться инвариантности относительно поворота, моменты вычисляются так, чтобы координаты x , y оставались внутри круга радиуса r .

Дескриптор BRIEF представляет собой битовую строку, описывающую часть изображения с помощью бинарных тестов интенсивности. Для каждой точки проводится тест. Если ее интенсивность меньше некоторого значения, то бит в строке, соответствующий этой точке, устанавливается нулю, и единице - в противном случае.

В работе [12] была предложена схема, которая позволяет описать регион изображения инвариантно относительно поворота.

2.2.2. HOG

HOG (Histogram of oriented gradients) представляет собой дескриптор характеристик, основанный на гистограмме ориентированных градиентов. Обычно этот дескриптор используется при обнаружении объектов. HOG вычисляется на густой сетке равномерно распределенных ячеек и для увеличения точности обнаружения использует нормализацию локального контраста.

Впервые HOG был описан в работе [13] в 2005 году на конференции «Computer Vision and Pattern Recognition» (CVPR). Основное внимание в ней было уделено обнаружению пешеходов в статичных изображениях.

Также были предприняты попытки применения данного дескриптора в задачах распознавания машин и животных. Поиск был произведен по форме объекта на разных масштабах и регионах интереса.

Внешний вид объекта и его форма могут быть представлены с помощью распределения интенсивности градиентов или направления углов. Изображение делится на мелкие регионы (ячейки), а каждый пиксель “описывается” внутри региона вычисляется с помощью гистограммы градиентов направления.

Дескриптор HOG имеет ряд достоинств по сравнению с другими. Он инвариантен к геометрическим и аффинным преобразованиям, за исключением поворота мелких объектов, однако чувствителен к поворотам всего изображения, всей сцены.

Реализация алгоритма HOG:

1. Вычисление градиента

Как и во многих детекторах, первым шагом перед вычислением вычисления градиента является предварительная обработка изображения. Однако в HOG можно игнорировать эту предобработку, принимая во внимание то, что нормализация гистограмм позволяет добиться похожих результатов.

2. Сортировка и создание гистограммы

Для каждого пикселя внутри региона интереса проводится взвешенное голосование по одной из величин гистограммы, основанной на вычислении градиентов.

Регионы интереса могут быть прямоугольными или круглыми. Значения гистограммы распределены между 0 до 180 или от 0 до 360.

3. Поблочные дескрипторы

Для того, чтобы посчитать изменения в контрасте или в освещении, значения градиентов должны быть нормализованы. Дескриптор HOG сворачивается в вектор, компоненты которого являются нормализованными гистограммами для каждого региона. Эти блоки могут пересекаться.

4. Нормализация блоков

Пусть v – вектор, содержащий все гистограммы данного блока $\|v\|_k$ это k -я норма $k = 1, 2$ и e – малая константа.

Нормализация проводится относительно нормы:

$$\text{Норма L2: } f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}}$$

$$\text{Корень квадратный из нормы L1: } f = \sqrt{\frac{v}{\|v\|_1 + e}}$$

После вычисления и нормализации вектора характеристик, готовый вектор подается на вход одному из классификаторов. Типичным вариантом классификатора является метод опорных векторов [15].

Обычно гистограмма ориентированных градиентов используется в распознавании людей (пешеходов), однако в работе [16] была предложена идея использования HOG для распознавания сцен, в которой вычисление гистограммы проводится на всём изображении, расширяя, таким образом, регион интереса до всего изображения.

В данной работе мы проверим эту гипотезу, обучив классификатор распознаванию сцен с помощью большой базы изображений, содержащей фотографии, сделанные как внутри помещения, так и вне.

3. ПРОГРАММНАЯ СИСТЕМА И ЭКСПЕРИМЕНТЫ

3.1. Программная система

Для моделирования системы используется язык Python и библиотеки OpenCV 2.4.9 и scikit-learn.

- Программа работает в следующих режимах:
- Вычисление характеристик данных изображений;
- Обучение модели по данной базе с характеристиками и классами;
- Проверка точности обученной модели на тестовой выборке изображений;
- Работа в реальном времени (видео).

Вычисление характеристик данных изображений

На вход подаются изображения вместе с метками классов.

Программа вычисляет характеристики изображений (HOG, ORB, GIST). Составляется словарь, в котором метки являются ключами, а значением – матрица, строки которой соответствуют вектору характеристик i -го изображения.

Составленный словарь сохраняется в файл.

Обучение модели

На вход подается выборка для обучения, состоящая из вектора характеристик и меток, соответствующих одному из двух классов. Программа использует одну из моделей обучения из библиотеки scikit-learn.

Обученная модель сохраняется в файл для дальнейшего использования.

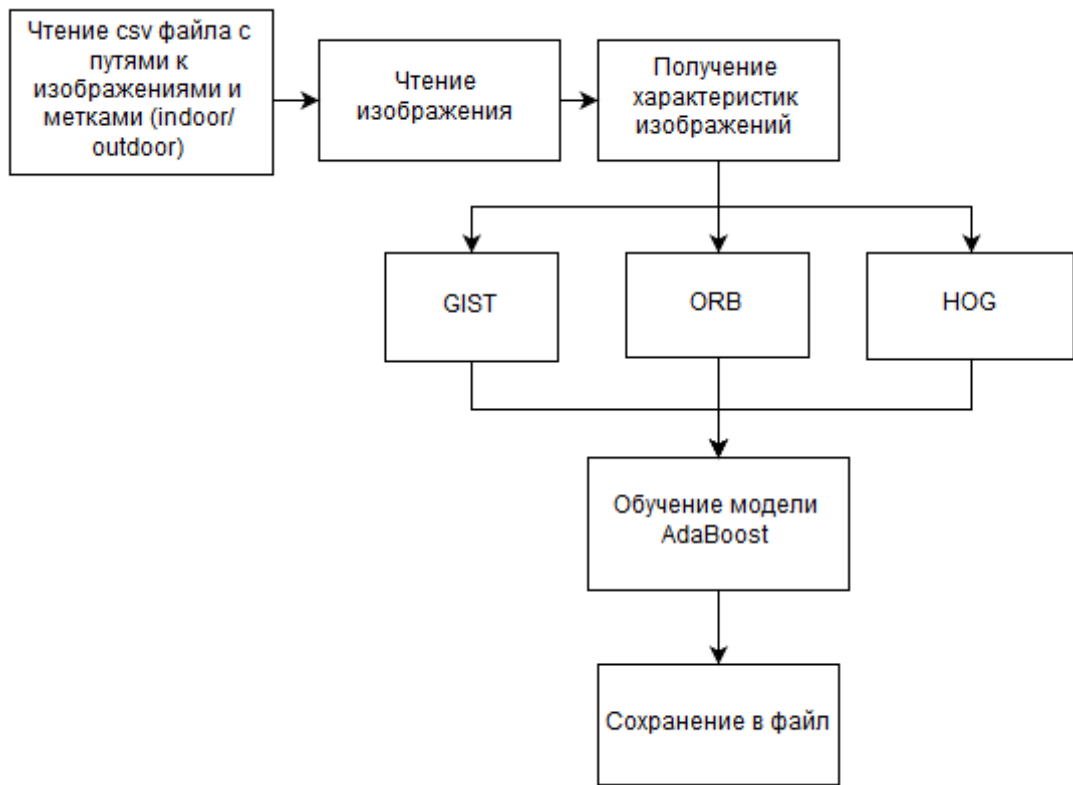


Рис. 2. Работа программы в режиме обучения

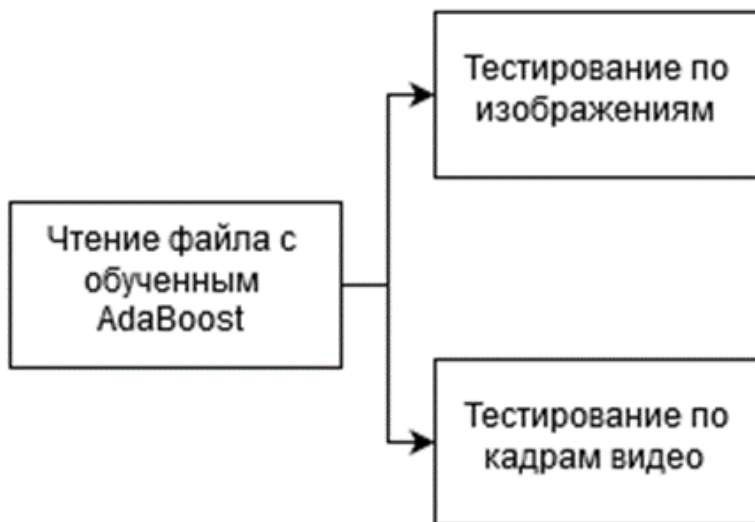


Рис. 3. Работа программы в режиме тестирования

Оценка модели

На вход подаётся обученная модель и вектор характеристик тестовой выборки с метками классов. Программа пытается предсказать класс данного изображения по вектору характеристик, а также сравнивает с истинным классом. Количество неправильно квалифицированных изображений делится на общее количество тестовых, таким образом получается процент неправильно классифицированных изображений.

Работа в реальном времени(видео)

На вход подается видео, полученное с веб-камеры. По данной обученной модели программа пытается предсказать класс каждого кадра. Для 25 кадров программа выдает самый часто встречающийся класс, а также коэффициент уверенности в нем.

$$P_C = \frac{1}{25} \sum_i p_{C_i},$$

где P_{in} – коэффициент уверенности в классе C для всех 25 кадров

p_{in_i} – коэффициент уверенности в классе C для i -го кадра

В итоге, классом для данных кадров будет класс с максимальным коэффициентом уверенности.

3.2. Эксперименты с базой изображений

3.2.1 База изображений

Для обучения и тестирования была использована база SUN [17], которая содержит 180 000 изображений. Каждому изображению соответствуют метки классов (число, представляющее класс). Все метки хранятся в csv файле.

Каждое изображение хранится в отдельной папке, название которой соответствует одной из 397 категорий (озеро, замок, пляж и т.д.).

База изображений SUN была создана с помощью проставления тегов классов изображений людьми. Теги ставились в два этапа. На первом этапе определялась одна из базовых категорий:

- 1) Фотография, сделанная внутри помещения;
- 2) Фотография, сделанная вне помещения на фоне природе;
- 3) Фотография, сделанная вне помещения на фоне объектов, созданных человеком.

Одной из проблем, с которой столкнулись создатели базы, оказалось содержание объектов как натуральных (деревья, кусты и т.д.), так и созданных человеком на одном изображении. Эта проблема была решена с помощью присвоения обеих меток.

На втором этапе ставились метки, расширяющие одну из базовых. Так, например, расширением для фотографий, сделанных внутри помещения, являются транспорт, дом, и т.д.

3.2.2. Проведенные эксперименты

Из базы данных случайно были выбраны 15452 изображения для обучения и 3857 изображений для проверки точности (тестирования).

Для сравнения мы используем самые популярные дескрипторы, применяемые к распознаванию сцен по представленным изображениям: GIST, HOG, ORB.

Мы используем реализацию GIST, представленную в [18].

Дескрипторы HOG и ORB вычисляются с помощью библиотек OpenCV

Для сравнения мы используем:

- время вычисления характеристик;
- точность классификации на тестовой выборке.

В таблице 1 приведены результаты обучения и тестирования на базе изображений SUN.

Таблица 1 – Результаты классификации базы данных SUN

	ORB	HOG	GIST
Точность	83%	93%	95%
Время обучения (сек.)	2400	2563	2632

Как мы видим, время обучения не так сильно отличается для различных характеристик. Однако, данное обучение проводилось с уже готовыми, вычисленными характеристиками.

Сравнение различных дескрипторов и времени вычисления для одного изображения в среднем приведено на рисунке 4.

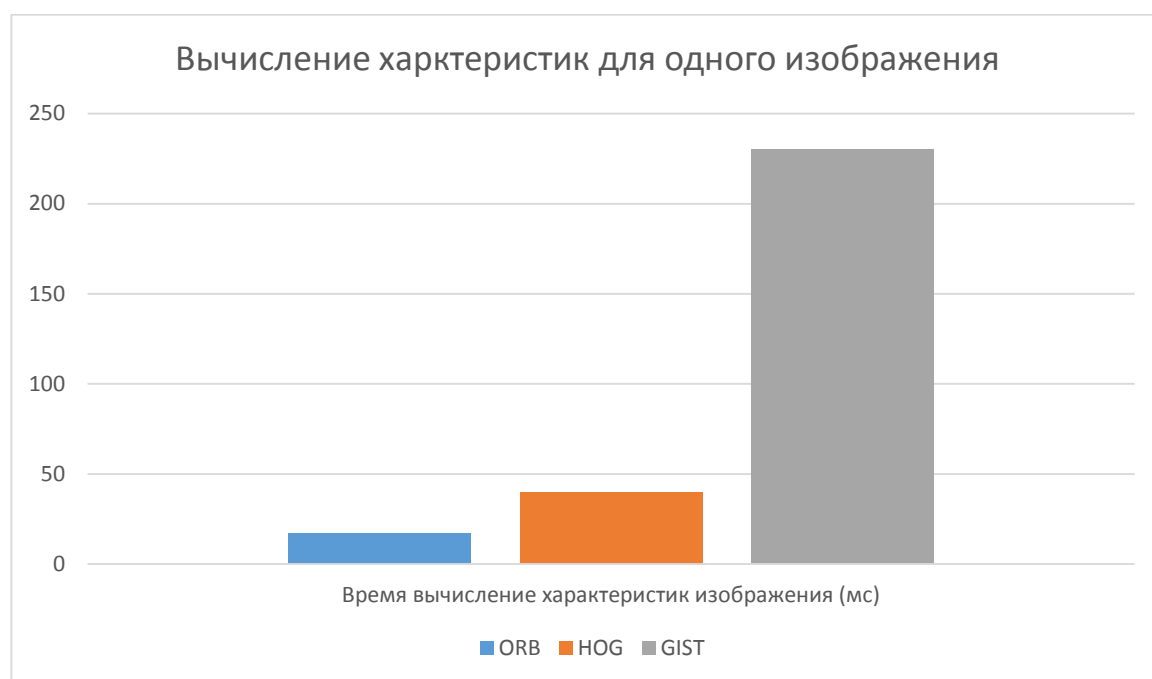


Рис. 4. Сравнение различных характеристик: ORB, HOG, GIST. Числа, представленные на диаграмме, соответствуют среднему времени вычисления характеристик для 18000 изображений.

Из диаграммы можно сделать вывод, что вычисление дескриптора GIST занимает значительно больше времени, чем для дескрипторов ORB или HOG.

Все изображения в базе данных имеют разные размеры, начиная с 300×200 и заканчивая 4500×3200 . Для того, чтобы характеристики имели одну длину, все изображения необходимо привести к одному размеру, например, к разрешению 30×30 . Как видно из диаграммы, наибольшее время тратится на прочтение и загрузку изображения в оперативную память.



Рис. 5. Сравнение времени для вычисления характеристик.

На диаграмме представлено среднее время, необходимое для чтения изображения с жёсткого диска, сжатие изображения и для вычисления характеристик HOG для сжатого изображения.

Поэтому, для того, чтобы ускорить чтение изображений, было решено провести эксперименты с видео. Видео, поступающее с веб-камеры, загружается непосредственно в память, не тратя таким образом время на чтение изображений с жесткого диска.

3.3. Эксперименты в реальном времени

Данные эксперименты направлены на проверку работы алгоритма классификации в режиме реального времени.

На рисунках 6 и 7 представлены результаты классификации внутри помещения и вне помещения. Как показали результаты экспериментов, применяя дескриптор Hog и классификатор AdaBoost, можно добиться классификации изображения или кадра видео в реальном времени и реагировать на движения и изменения в глобальной сцене изображения.



Рис. 6. Результат классификации в реальном времени вне помещения.

Данная программа выдает коэффициент уверенности в результате классификации от 0.5 - 1.0. В итоге, все изображения делятся на два класса. Если коэффициент уверенности меньше чем 0.5 для класса внутри помещения, то это означает, что коэффициент уверенности будет больше 0.5 для класса

вне помещения. Программа переключит контекст и выдаст другой ответ, соответствующий классу с наибольшим коэффициентом уверенности.



Рис. 7. Результат классификации внутри помещения.

ЗАКЛЮЧЕНИЕ

В данной работе была рассмотрена проблема распознавания сцен по представленным изображениям. Был предложен алгоритм быстрой классификации сцен, который показал высокую точность хорошие результаты, и также была продемонстрирована возможность применения дескриптора HOG в качестве быстрого дескриптора для классификации сцен.

Дескрипторы, позволяющие быстро получить характеристики изображения, могут быть использованы для обработки в реальном времени. Одной из проблем, с которой мы столкнулись во время обработки базы изображений, являлось неэффективное чтение изображений с жесткого диска. 90 процентов времени было потрачено на чтение и загрузку в оперативную память изображения. Данная проблема была решена получением изображений с помощью веб-камеры прямой загрузкой кадров видео, уменьшая таким образом общее время обработки изображений. Вся обработка свелась к вычислению характеристик и сжатию изображений, что выполняется менее чем за 20 мс.

Данный алгоритм в дальнейшем может быть применен на большее число классов. Так, например, после распознавания базовой сцены (вне помещения, внутри помещения), можно запустить распознавание подклассов – офис, дом, замок, пляж и т.д.

СПИСОК ЛИТЕРАТУРЫ

1. Moreels P., Perona P. Evaluation of features detectors and descriptors based on 3d objects //International Journal of Computer Vision. – 2007. – Т. 73. – №. 3. – С. 263-284.
2. Juila Vogel and Bernt Schiele, “A semantic typicality measure for natural scene categorization,” Pattern Recognition Symposium DAGM, 2004.”
3. Tversky, B.,Hemenway, K.: Categories of environmental scenes. Cognitive Psychology 15 (1983).
4. Rogowitz, B., Frese, T., Smith, J., Bouman, C., Kalin, E.: Perceptual image similarity experiments. In: SPIE Conf. Human Vision and Electronic Imaging, San Jose, California (1998).
5. Gorkani M. M., Picard R. W. Texture orientation for sorting photos" at a glance" //Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision; Image Processing., Proceedings of the 12th IAPR International Conference on. – IEEE, 1994. – Т. 1. – С. 459-464.
6. Bosch A., Zisserman A., Muoz X. Scene classification using a hybrid generative/discriminative approach //Pattern Analysis and Machine Intelligence, IEEE Transactions on. – 2008. – Т. 30. – №. 4. – С. 712-727.
7. Oliva A., Torralba A. Modeling the shape of the scene: A holistic representation of the spatial envelope //International journal of computer vision. – 2001. – Т. 42. – №. 3. – С. 145-175.
8. Mehrotra R., Namuduri K. R., Ranganathan N. Gabor filter-based edge detection //Pattern Recognition. – 1992. – Т. 25. – №. 12. – С. 1479-1494.
9. Воронцов К. В. Лекции по методу опорных векторов //ВЦ РАН, М.: URL: <http://www.ccas.ru/voron/download/SVM.pdf> (дата обращения: 03.03. 12). – 2007.
10. T. Hastie, R. Tibshirani and J. Friedman, “Elements of Statistical Learning Ed. 2”, Springer, 2009.

- 11.** Zhu, H. Zou, S. Rosset, T. Hastie, “Multi-class AdaBoost”, 2009.
- 12.** Rublee E., Rabaud V., Konolige K., Bradski G. ORB: an efficient alternative to SIFT or SURF // In Proceedings of the International Conference on Computer Vision (CVPR’11). 2011. P. 2564-2571.
- 13.** Dalal N., Triggs B. Histograms of oriented gradients for human detection //Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. – IEEE, 2005. – T. 1. – C. 886-893.
- 14.** Tuzel O., Porikli F., Meer P. Region covariance: A fast descriptor for detection and classification //Computer Vision–ECCV 2006. – Springer Berlin Heidelberg, 2006. – C. 589-600.
- 15.** Dalal N., Triggs B. Histograms of oriented gradients for human detection //Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. – IEEE, 2005. – T. 1. – C. 886-893.
- 16.** Villamizar M. et al. Combining color-based invariant gradient detector with HoG descriptors for robust image detection in scenes under cast shadows //Robotics and Automation, 2009. ICRA'09. IEEE International Conference on. – IEEE, 2009. – C. 1997-2002.
- 17.** Xiao J. et al. Sun database: Large-scale scene recognition from abbey to zoo //Computer vision and pattern recognition (CVPR), 2010 IEEE conference on. – IEEE, 2010. – C. 3485-3492.
- 18.** Oliva A., Torralba A. Building the gist of a scene: The role of global image features in recognition //Progress in brain research. – 2006. – T. 155. – C. 23-36.