электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

УДК 004.023

Методы и алгоритмы машинного обучения в задачах распознавания образов

***Гурьянов А.В.**, студент*
Россия, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Компьютерные системы и сети»

Научный руководитель: Андреев А.М., к.т.н., доцент
Россия, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Компьютерные системы и сети»
v.suzev@bmstu.ru

Машинное обучение – обширный подраздел искусственного интеллекта, изучающий методы построения моделей, способных обучаться, и алгоритмов для их построения и обучения [1].

Необходимость применения, решаемые задачи

К применению машинного обучения в области компьютерного зрения обратились впервые тогда, когда стало понятно, что уже существующие методы, такие как, например, поиск объектов по шаблону или распознавание по особым признакам, не могут решить задачу классификации объектов на изображении в целом. Трудности применения подобных методов в первую очередь возникали из-за того, что любой алгоритм распознавания приходилось составлять вручную, что приводило к перебору большого количества вариантов классификации. Более того, возможно было лишь проводить оценку по осмысленным признакам (цвет, форма объекта), что не давало возможности стабильно распознавать объекты в различных схемах освещения и в состоянии легкой деформации.

Использование методов машинного обучения позволяет строить более сложные и совершенные алгоритмы распознавания на основе большого количества собранных данных.

Для решения этой задачи мы располагаем следующей информацией:

- Некоторое количество данных, называемое обучающей выборкой.
- Каждый элемент выборки описывается набором признаков **x,** называемым вектор-признак.

- Для каждого элемента выборки известен ответ **y** о принадлежности его к одному из классов.

Машинное обучение помогает решить следующие варианты задач:

- Бинарная классификация.
- Многоклассовая классификация.
- Регрессия.
- Кластеризация.

Решение каждой задачи предполагает ее тщательное изучение, выбор наилучшего метода или набора методов и подходов к их реализации. Рассмотрим некоторые методы, которые применяются для решения большинства типов задач.

Метод опорных векторов (Support Vector Machine, SVM)

Этот метод является одним из ключевых в сфере компьютерного зрения. Более того, он дал толчок для развития в области анализа данных.

Идея метода заключается в следующем: имеется обучающая выборка вида

$$(\boldsymbol{x_1}, y_1), (\boldsymbol{x_2}, y_2), \ldots, (\boldsymbol{x_n}, y_n),$$

где $\boldsymbol{x_i}$ – объект выборки, представляющий собой вещественный вектор, y – параметр объекта, определяющий, к какому классу он принадлежит.

Необходимо найти линейную функцию, разделяющую точки, принадлежащие одному классу $\{y = +1\}$ и точки, принадлежащие другому классу $\{y = -1\}$.

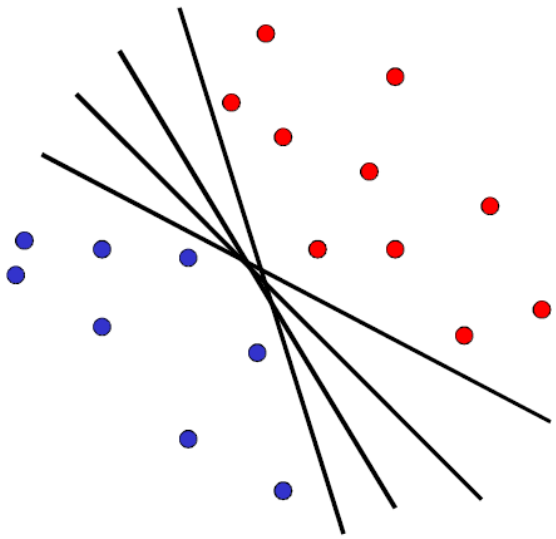

Рис. 1. Линейная классификация

Очевидно, что в данной ситуации можно построить множество таких плоскостей. Однако необходимо выбрать одну. С точки зрения точности классификации лучшим решением будет выбор прямой, максимизирующей расстояние между ближайшими объектами разных классов (сплошная прямая на рисунке 2). При классификации объектов в пространстве большей размерности такая прямая называется оптимальной разделяющей гиперплоскостью [2].

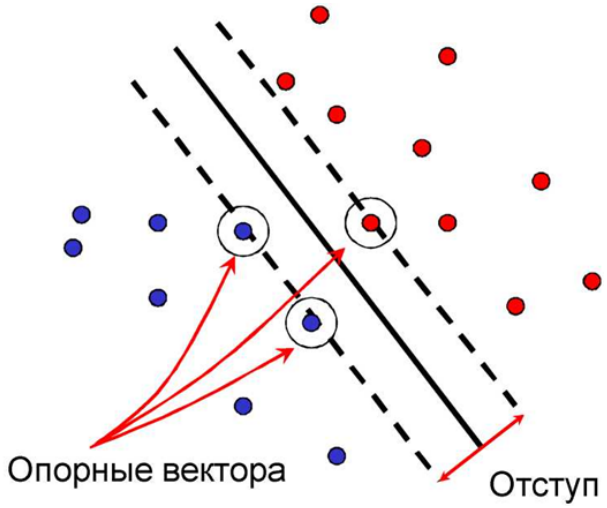

Рис. 2. Оптимальная разделяющая гиперплоскость

Объекты классификации (в нашем случае – точки), наименее удаленные от построенной прямой называются опорными векторами. Параметры прямой (гиперплоскости) мы можем задать таким образом, что будут выполняться следующие неравенства:

для гиперплоскости: $\boldsymbol{x_i} \cdot \mathbf{w} + b = 0,$

для (+) $\mathbf{x_i}$: $\boldsymbol{x_i} \cdot \mathbf{w} + b \geq 1$

для (–) $\mathbf{x_i}$: $\boldsymbol{x_i} \cdot \mathbf{w} + b \leq -1$, где $\mathbf{w}$ – нормальный вектор к разделяющей гиперплоскости, b — вспомогательный параметр (расстояние от гиперплоскости до начала координат).

Тогда для опорных векторов будет справедливо равенство:

$$\boldsymbol{x_i} \cdot \mathbf{w} + \mathrm{b} = \pm 1.$$

Исходя из полученных выражений, делаем вывод, что формула для расчета расстояния от любой точки до гиперплоскости имеет следующий вид:

$$\frac{\boldsymbol{x_i} \cdot \mathbf{w} + \mathrm{b}}{\|\mathrm{w}\|}$$

То есть расстояние от ближайших точек (опорных векторов) до гиперплоскости будет равно $\frac{1}{\|w\|}$ и, соответственно, расстояние между ними $\frac{2}{\|w\|}$.

Для нахождения наибольшего значения отступа необходимо найти максимальное значение для $\frac{2}{\|w\|}$ (или минимальное для $\|w\|$). Такая задача называется задачей квадратичной оптимизации:

$$\begin{cases} \min(\|w\|^2) \\ y_i(\boldsymbol{x_i} \cdot \mathbf{w} + b) \geq 1 \end{cases}$$

В процессе решения, находится значение параметра **w**. Ответ будет представлен в виде:

$$\mathbf{w} = \sum \alpha_i \, y_i \boldsymbol{x_i} \, ,$$

где a_i – некоторый параметр (если $a_i \neq 0$, то $\boldsymbol{x_i}$ - опорный вектор)

В результате, окончательная решающая функция будет иметь вид:

$$\boldsymbol{x} \cdot \mathbf{w} + b = \boldsymbol{x} \cdot \sum \alpha_i \, y_i \boldsymbol{x_i} + b,$$

По виду данной формулы можно сделать вывод о том, что решение о принадлежности объекта $\boldsymbol{x}$ к интересующему нас классу зависит от суммы векторных произведений этого объекта с каждым из опорных векторов.

В реальности же, идеального разбиения на 2 класса достичь не удается. Для любой гиперплоскости по обе стороны всегда будут находиться данные обоих классов (рисунок 3).

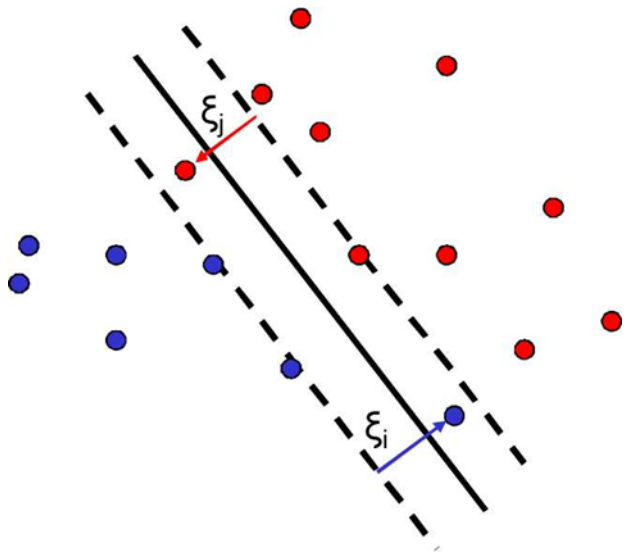

Рис. 3. Классификация в реальном случае

Поэтому в решение вводятся $\xi = (\xi_1, .., \xi_n)^T$ – «slack» переменные, и задачей становится нахождение прямой, максимизирующей отступ при минимальном количестве ложно классифицированных точек:

$$\begin{cases} \min(\|w\|^2 + \sum_{i=1}^{n} \xi i) \\ y_i(\boldsymbol{x_i} \cdot \mathbf{w} + b) \geq 1 - \xi i \end{cases}$$

SVM прекрасно работает на линейно-разделимых данных (рисунок 4).

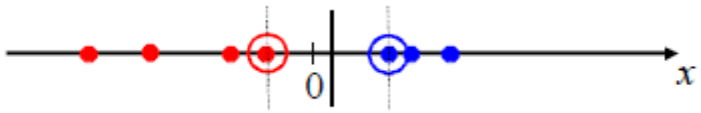

Рис. 4. Линейно разделимые данные

Однако обычно данные изначально линейно неразделимы (рисунок 5).

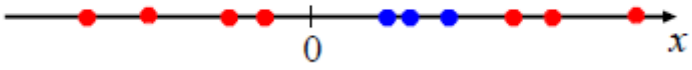

Рис. 5. Линейно неразделимые данные

В таких случаях появляется необходимость отобразить их на пространство большей размерности так, чтобы они стали линейно-разделимыми в новой системе координат (рисунок 6).

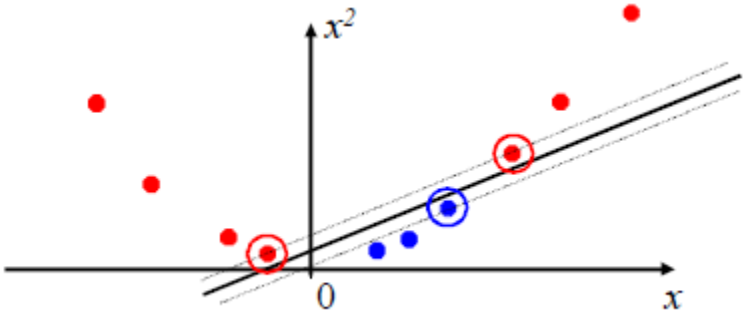

Рис. 6. Переход к двумерной системе координат

В связи со сложностью подобного отражения при переходе к пространствам большой размерности, в методе опорных векторов используются ядровые функции. Суть данного приема заключается в том, что вместо прямого вычисления сложного преобразования $\varphi(x)$, мы выбираем функцию $K(x_i , x_{jj})$, такую что:

$$\boldsymbol{K(x_i , x_{jj}) = \varphi(x_i) \cdot \varphi(x_j)} .$$

При этом решающая функция будет иметь следующий вид:

$$\sum \alpha_i \, y_i \boldsymbol{K}(\boldsymbol{x_i}, \boldsymbol{x}) + b.$$

Главные достоинства метода:

- Много доступных библиотек.
- Возможность отображения в пространство большей размерности с использованием ядровых функций.
- Хорошая точность классификации, даже на небольших обучающих выборках.

Недостатки:

- Нет прямых многоклассовых методов;
- Использование ядровых функций приводит к большому количеству вычислений;

Алгоритм k-средних (k-means)

Данный алгоритм, а на настоящий момент его более совершенные модификации, чаще всего применяются для решения задач кластеризации, то есть разбиения объектов выборки на сравнительно однородные группы.

Множество элементов векторного пространства разбивается на заранее известное число кластеров k. Действие алгоритма таково, что он стремится минимизировать сумму квадратов евклидовых расстояний между точками x_i и ближайшими центрами кластеров m_k:

$$D(x, m) = \sum_{cluster\ k} \sum_{piont\ i\ in\ cluster\ k} (x_i - m_k)^2.$$

Основная идея метода заключается в том, что на каждой итерации вычисляется центр масс для каждого кластера, полученного на предыдущем шаге [3]. Затем векторы разбиваются на кластеры вновь в соответствии с тем, какой из новых центров оказался ближе. Алгоритм завершается, когда на какой-то итерации не происходит изменения кластеров (рисунок 7).

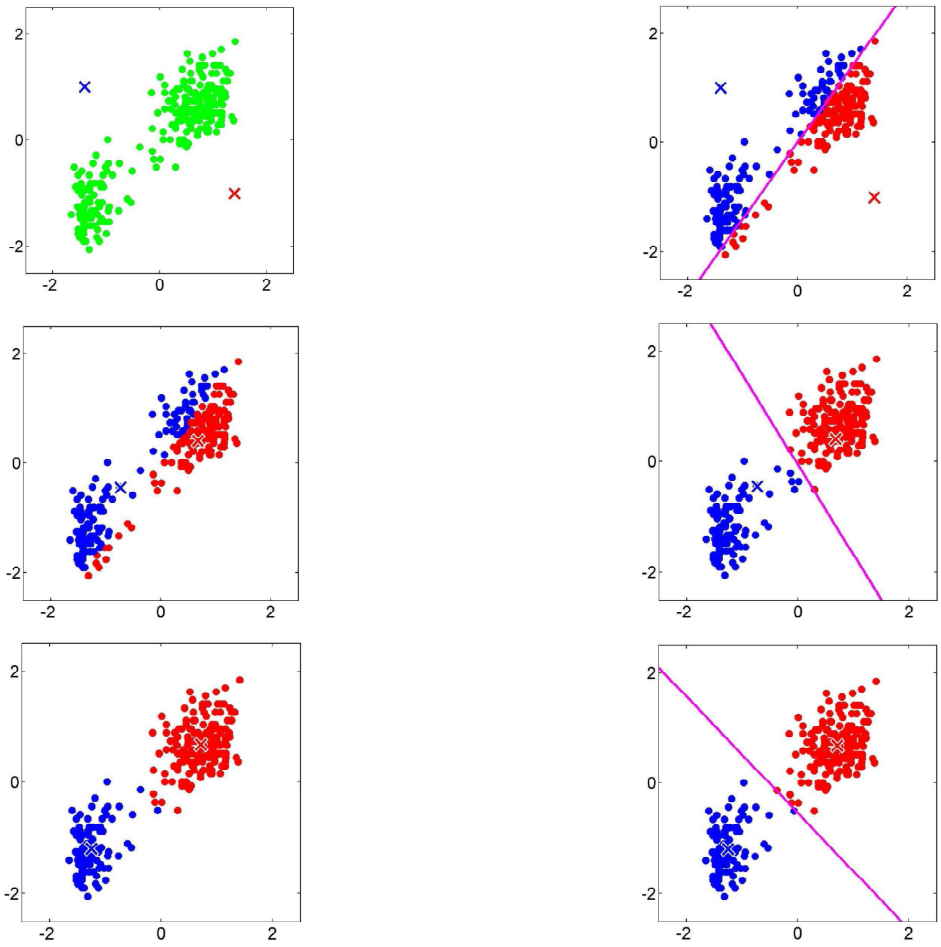

Рис. 7. Кластеризация методом К-средних

Главным достоинством алгоритма является простота его реализации. В то же время его применение сопровождается следующими проблемами:

- Необходимо заранее знать количество кластеров.
- Алгоритм очень чувствителен к выбору начальных центров кластеров (в классическом варианте подразумевается случайный выбор, что иногда вызывает серьезные погрешности, поэтому желательно проводить исследования объекта для более точного определения начальных центров).

Бустинг (boosting)

Бустинг – комплекс методов, способствующих повышению точности аналитических моделей. В концепции машинного обучения под бустингом понимают процедуру последовательного построения композиции алгоритмов с расчетом того, что каждый следующий алгоритм стремится компенсировать недостатки композиции всех предыдущих [4].

Одним из лучших на сегодняшний день алгоритмов усиления классификаторов является AdaBoost.

Принцип работы AdaBoost можно представить следующей последовательностью действий:

- Обучается простой классификатор.
- Рассчитывается вес (степень влияния на окончательное решение) классификатора (вес обратно пропорционален вероятности ошибки классификатора).
- Выполняется перерасчет весов примеров из выборки, чтобы в дальнейшем правильно распознать те примеры, которые были неверно распознаны на данном этапе (если пример классифицировался правильно – вес уменьшается, если неправильно – вес увеличивается).
- Алгоритм повторяется n раз. Линейная сумма классификаторов образует итоговый классификатор.

Математически выражение для итогового классификатора записывается так:

$$H(x) = \text{sign}\left(\sum_{k=1}^{k} \alpha_k h_k(x)\right),$$

где $h_k(x)$ – k-й простой классификатор, α_k – его вес, sign – оператор линейной суммы.

Несколько промежуточных этапов построения сложного классификатора из линейных с использованием AdaBoost представлены на рисунке 8. На каждом этапе желтым цветом отмечена область, в которую попали точки, распознанные как принадлежащие классу синих точек, а зеленым – точки, не принадлежащие данному классу.

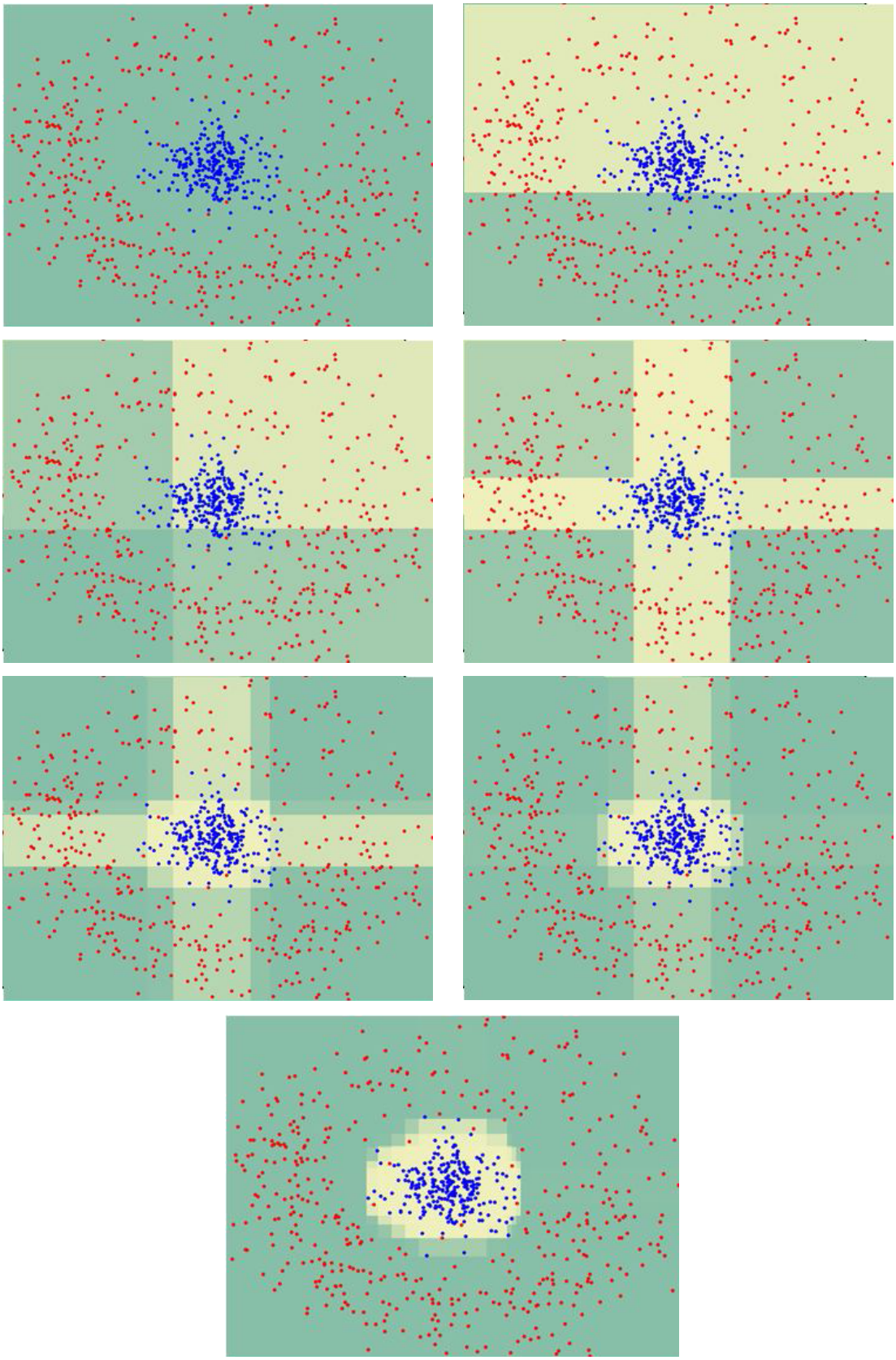

Рис. 8. Построение классификатора с AdaBoost

Достоинства алгоритма:

- Универсальность.

- Высокая скорость сходимости.
- Возможность очень эффективной программной реализации и распараллеливания.
- Простота метода.

Недостатки:

- Трудность определения нужного числа итераций обучения (зачастую, ошибка на контрольной выборке продолжает падать и после нулевой ошибки на обучающей выборке).
- Требует достаточно длинных обучающих выборок.

Выводы

В заключение стоит отметить, что практически любые современные системы различных областей компьютерного зрения, предполагающие использование технологий обработки большого количества данных и машинного обучения, используют как минимум один из представленных методов. Например, на основе AdaBoost построен детектор Виолы-Джонса – на сегодняшний день один из лучших методов распознавания лиц, который реализован во всех современных фотоаппаратах и ведеокамерах. Кроме того, очень популярный алгоритм распознавания под названием «Мешок слов», реализуется с использованием различных модификаций алгоритмов k-means и SVM.

Если же говорить о системах отслеживания объекта на видео, то одним из наиболее точных и быстрых является метод, основанный на применении разных классификаторов, из которых как минимум один использует метод опорных векторов с некоторым ядром, а другой строится на основе бустинга простых классификаторов. При этом для каждого кадра видео выбирается классификатор с лучшими показателями ошибки (false positive rate) и вероятности обнаружения (detection rate) [5].

Подводя итоги, необходимо сказать, что при подходе к решению задачи разработки и проектирования серьезной системы в области компьютерного зрения, предполагающей предварительное обучение на больших объемах изображений или видеоданных, в первую очередь необходимо обратиться к рассмотренным в данной статье алгоритмам и изучить варианты их реализации на основе уже существующих методов. Такой подход не только поможет выбрать наиболее удачный для конкретной задачи метод, но и позволит найти пути к его возможному улучшению.

Список литературы

1. Машинное обучение. Режим доступа: http://ru.vlab.wikia.com/wiki/Машинное_обучение (дата обращения 17.10.2014).
2. Лепский А. Е., Броневич А. Г. Математические методы распознавания образов: Курс лекций. Таганрог: Изд-во ТТИ ЮФУ, 2009. 155 с.
3. Кластеризация: алгоритмы k-means и c-means. Режим доступа: http://habrahabr.ru/post/67078/ (дата обращения 23.10.2014).
4. Метод Виолы-Джонса (Viola-Jones) как основа для распознавания лиц. Режим доступа: http://habrahabr.ru/post/133826/ (дата обращения 02.11.2014).
5. Введение в компьютерное зрение. Режим доступа: https://www.lektorium.tv/course/22847 (дата обращения 10.10.2014).