

# Описание программы «Распознавание жестов»

Программа написана на java с использованием openCV.

Процесс установки openCV и примеры его использования могут быть найдены на [http://docs.opencv.org/2.4/doc/tutorials/introduction/desktop\\_java/java\\_dev\\_intro.html](http://docs.opencv.org/2.4/doc/tutorials/introduction/desktop_java/java_dev_intro.html)

Текущая версия кода находится на <https://github.com/BOBAHDEP/GesturesRecognition>

## 1. GUI

Программа состоит из одного главного окна..

Оно отображает видео, полученное с веб камеры. На этих кадрах прямоугольниками обводятся распознанные жесты. Рядом с прямоугольником отображается название типа жеста. В этом окне отображаются все найденные на видео жесты. Один из них считается наиболее достоверным и отмечается розовым прямоугольником. Если в кадре распознаны другие жесты, они обводятся белыми прямоугольниками.

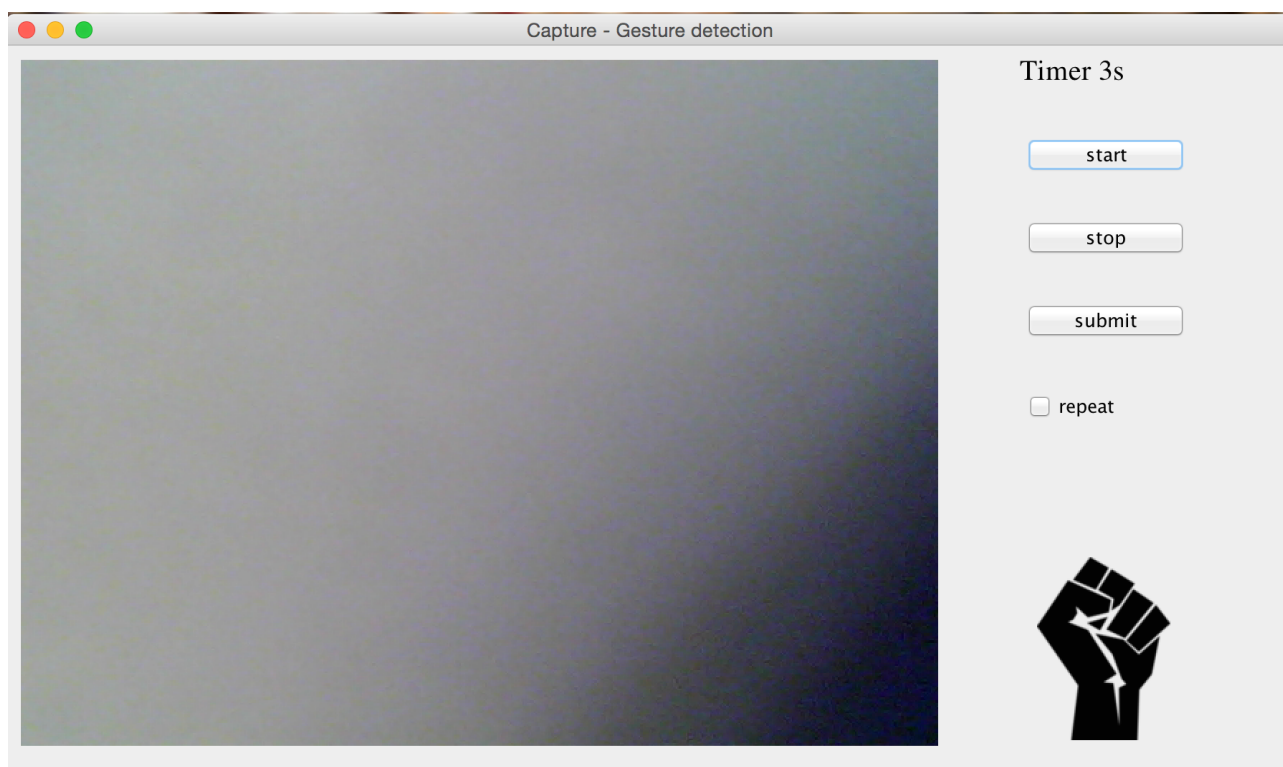


Рис. 1 Пример главного окна

Также отображаются картинки или надписи с инструкциями: приготовиться к распознаванию жестов, картинка с жестом, который система распознает, сообщение при удачном распознавании правильного жеста.

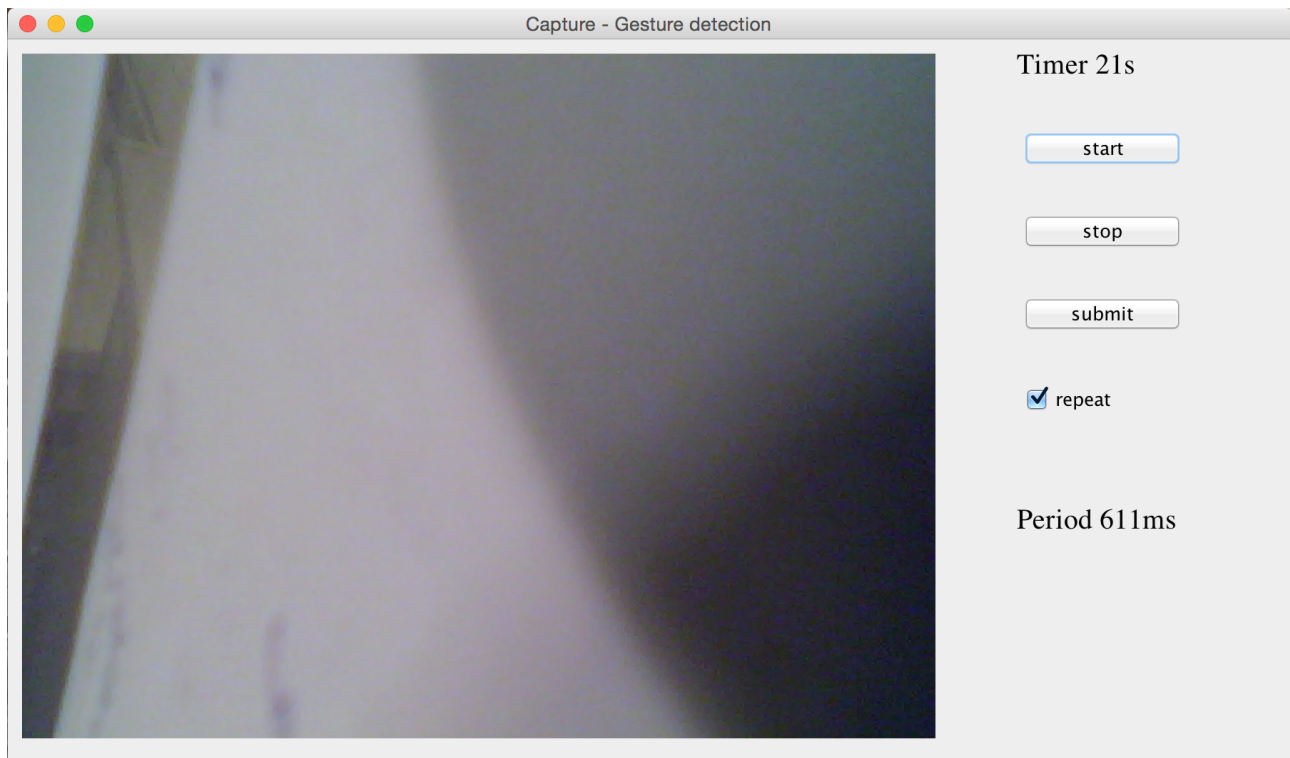
Управление событиями происходит при нажатии клавиш клавиатуры:

пробел - установить дефолтное состояние, надпись «приготовьтесь»

буква, совпадающая с первой буквой названия жеста (например, «F» - fist)

запускает процесс распознавания соответствующего жеста, показывается картинка, иллюстрирующая его.

В программе реализовано 2 режима: повторить жест и повторить последовательность жестов. Во втором случае считается период между распознанными пробами.



Все данные о результатах распознавания пишутся в лог - файл .log в следующем формате:  
DetectedFigure{x=139, y=21, width=190, height=190, figureType='palm'}  
java.util.GregorianCalendar[time=1473797424656,areFieldsSet=true,areAllFieldsSet=true,lenient=true,zone=sun.util.calendar.ZoneInfo[id="Asia/Riyadh",offset=10800000,dstSavings=0,useDaylight=false,transitions=3,lastRule=null],firstDayOfWeek=2,minimalDaysInFirstWeek=1,ERA=1,YEAR=2016,MONTH=8,WEEK\_OF\_YEAR=38,WEEK\_OF\_MONTH=3,DAY\_OF\_MONTH=13,DAY\_OF\_YEAR=257,DAY\_OF\_WEEK=3,DAY\_OF\_WEEK\_IN\_MONTH=2,AM\_PM=1,HOUR=11,HOUR\_OF\_DAY=23,MINUTE=10,SECOND=24,MILLISECOND=656,ZONE\_OFFSET=10800000,DST\_OFFSET=0].  
Фиксируется положение жеста, размер распознанной части, точное время и тип жеста. Дальше эти логи можно анализировать как угодно.

## 2. Алгоритм распознавания правильного жеста

С помощью машинного обучения (см. часть 3) на каждом кадре видео находятся распознанные жесты. Самый большой по размеру из них выбирается верным. Это условие обусловлено большим количеством небольших неверно распознанных жестов, выявленных при тестировании. Оно еще будет корректироваться при доработке классификатора.

Кадры считываются по таймеру через определенный промежуток времени (для тестирования - 1 секунда, можно изменять).

Условия правильного считанного жеста:

на нескольких кадрах подряд (для тестирования 6, можно изменять) должен быть распознан жест в одном и том же месте (с определенной погрешностью), такого же размера (с определенной погрешностью) и одинакового типа, совпадающего с выбранным в начале нажатием клавиши клавиатуры.

## 3. Метод Виолы-Джонса для распознавания образов

Основные принципы, на которых основан метод, таковы:

- используются изображения в интегральном представлении, что позволяет вычислять быстро необходимые объекты;
- используются признаки Хаара, с помощью которых происходит поиск нужного объекта (в данном контексте, лица и его черт);
- используется бустинг (от англ. boost – улучшение, усиление) для выбора наиболее подходящих признаков для искомого объекта на данной части изображения;
- все признаки поступают на вход классификатора, который даёт результат «верно» либо «ложь»;
- используются каскады признаков для быстрого отбрасывания окон, где не найдено нужное изображение.

Обучение классификаторов идет очень медленно, но результаты поиска лица очень быстры, именно поэтому был выбран данный метод распознавания лиц на изображении. Виола-Джонс является одним из лучших по соотношению показателей эффективности распознавания/скорость работы. Также этот детектор обладает крайне низкой вероятностью ложного обнаружения лица.

Принцип сканирующего окна

В общем виде, задача обнаружения образов на цифровом изображении выглядит именно так:

- имеется *изображение*, на котором *есть искомые объекты*. Оно представлено *двумерной матрицей пикселей размером  $w \times h$* , в которой каждый пиксель имеет значение:
- — от 0 до 255, если это черно-белое изображение;
- — от 0 до 255<sup>3</sup>, если это цветное изображение (компоненты R, G, B).
- в результате своей работы, алгоритм должен определить жесты и *пометить их* – поиск осуществляется в *активной области* изображения *прямоугольными признаками*, с помощью которых и описывается найденный образ

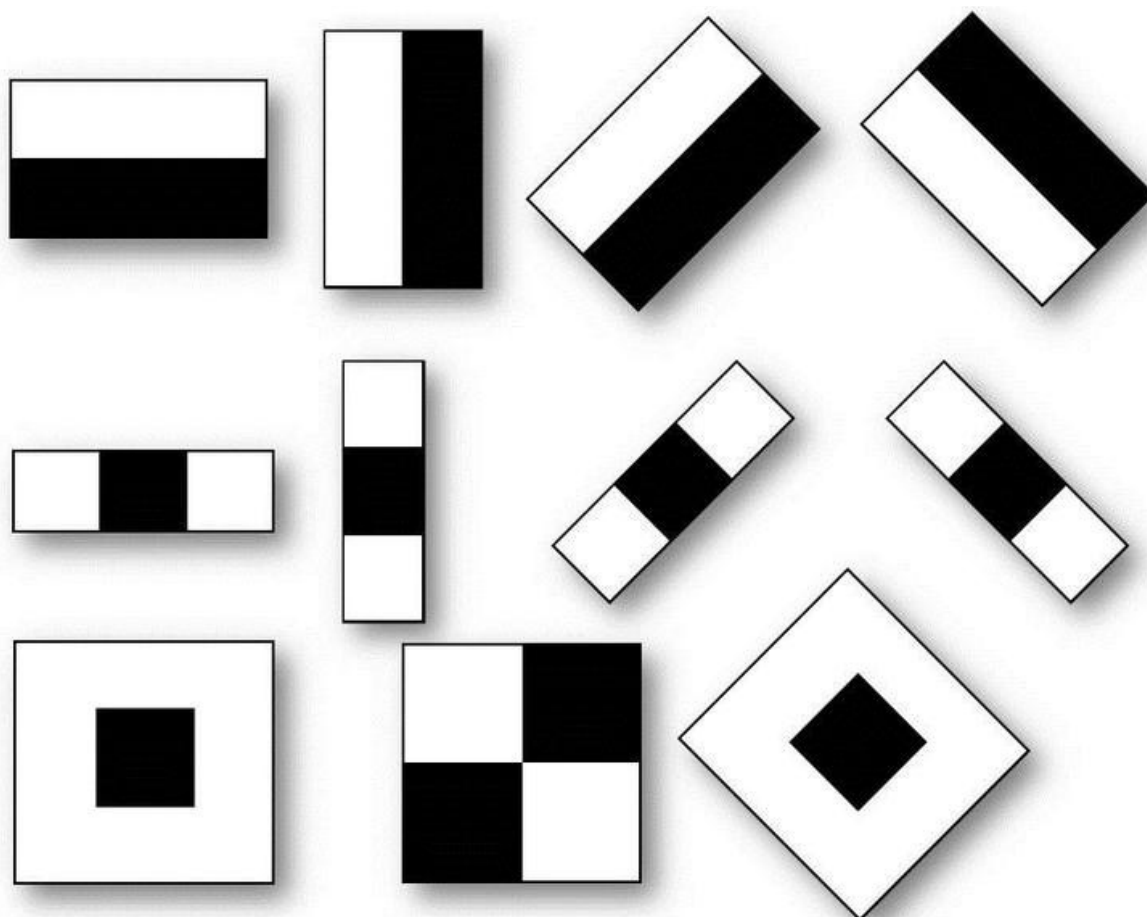
Интегральное представление изображений

Для того, чтобы производить какие-либо действия с данными, используется *интегральное представление* изображений в методе Виолы-Джонса. Такое представление используется часто и в других методах, к примеру, в вейвлет-преобразованиях, SURF и многих других разобранных алгоритмах. Интегральное представление позволяет быстро рассчитывать *суммарную яркость* произвольного прямоугольника на данном изображении, причем какой бы прямоугольник не был, время расчета неизменно.

Интегральное представление изображения – это матрица, совпадающая по размерам с исходным изображением. В каждом элементе ее хранится сумма интенсивностей всех пикселей, находящихся левее и выше данного элемента.

Признаки Хаара

*Признак* — отображение  $f: X \Rightarrow D_f$ , где  $D_f$  — множество допустимых значений признака. В стандартном методе Виолы – Джонса используются прямоугольные признаки, изображенные на рисунке ниже, они называются *примитивами Хаара*:



Применяемый в алгоритме бустинг, AdaBoost

Для решения задачи обучения существует технология *бустинга*.

Бустинг — комплекс методов, способствующих повышению точности аналитических моделей. *Эффективная модель, допускающая мало ошибок классификации, называется «сильной».* «Слабая» же, напротив, *не позволяет надежно разделять классы или давать точные предсказания, делает в работе большое количество ошибок.* Поэтому **бустинг** (от англ. boosting – повышение, усиление, улучшение) означает дословно **«усиление» «слабых» моделей** – это процедура последовательного построения композиции алгоритмов машинного обучения, когда каждый следующий алгоритм стремится компенсировать недостатки композиции всех предыдущих алгоритмов.

Идея бустинга была предложена Робертом Шапиром (*Schapire*) в конце 90-х годов, когда надо было найти решение вопроса о том, чтобы имея множество плохих (незначительно отличающихся от случайных) алгоритмов обучения, получить один хороший. В основе такой идеи лежит построение *цепочки (ансамбля) классификаторов*, который называется **каскадом**, каждый из которых (кроме первого) *обучается на ошибках предыдущего*. Например, один из первых алгоритмов бустинга Boost1 использовал каскад из 3-х моделей, первая из которых обучалась на всем наборе данных, вторая – на выборке примеров, в половине из которых первая дала правильные ответы, а третья — на примерах, где «ответы» первых двух разошлись. Таким образом, имеет место последовательная обработка примеров каскадом классификаторов, причем так, что задача для каждого последующего становится труднее. Результат определяется путем простого голосования: пример относится к тому классу, который выдан большинством моделей каскада.

Для обучения используются встроенные утилиты `opencv`.

4. Ссылки:

<https://habrahabr.ru/post/133826/> - описание метода распознавания

