

Санкт-Петербургский Государственный Университет
Математико-механический факультет
Кафедра системного программирования

Землянская Светлана Андреевна

**Технология распознавания жестов по видео с использованием
методов машинного обучения**

Дипломная работа

Допущена к защите.

Заведующий кафедрой:

д.ф.-м.н., профессор А.Н. Терехов

Научный руководитель:

д.ф.-м.н., профессор О.Н. Граничин

Рецензент:

аспирант Д.С. Кривоконь

Санкт-Петербург

2013

Saint Peterburg State University
Mathematics and Mechanics Faculty
System programming department

Zemlyanskaya Svetlana

**Technology Of Gestures Recognition On Video Stream Using
Machine Learning Algorithms**

Graduation Thesis

Admitted for defence.

Head of the chair:

Professor A.N. Terehov

Scientific supervisor:

Professor O.N. Granichin

Reviewer:

Postgraduate D.S. Krivokon

Saint Petersburg

2013

Содержание

1	Введение	3
2	Постановка задачи	5
3	Обзор существующих решений	7
4	Используемые методы	9
4.1	Вычитание фона на изображении	9
4.2	Гистограмма направленных градиентов	11
4.3	Машинное обучение	12
4.3.1	Постановка задачи классификации	12
4.3.2	Логистическая регрессия	12
4.3.3	Support Vector Machines	13
4.3.4	Нелинейная SVM	14
4.3.5	Мультиклассификатор	14
5	Алгоритм	15
6	Реализация	21
6.1	Проектирование	21
6.2	Особенности реализации	22
6.3	Тестирование	24
7	Результат работы	25

8 Заключение	27
8.1 Дальнейшее развитие	27
Литература	28

Глава 1

Введение

Уже более полувека развитие технологий приводит к тому, что нижняя граница размеров процессоров уменьшается, в то время как их производительность продолжает увеличиваться. Видимым рядовому пользователю результатом этого является разнообразие созданных интеллектуальных систем, используемых в повседневной жизни: от смартфонов до планшетов, от бытовой техники до домашних роботов. Камнем преткновения становится обмен информацией между компьютером и пользователем, растёт потребность в исследовании новых способов, более естественных, чем ввод данных с клавиатуры, эмулирующих бытовое общение между людьми. Активно развивающимся направлением решения проблемы является управление компьютером с помощью голосовых команд. Тем не менее, несмотря на значительные успехи в области, этот способ применим далеко не во всех ситуациях. Другим исследуемым способом ввода данных является использование визуальных систем: передача информации посредством мимики и жестов.

В работе было произведено исследование второго метода ввода информации - через жесты человека. В ходе исследования был разработан и реализован алгоритм по распознаванию жестов получаемых с видео пото-

ка. Важным требованием было минимизировать ограничения на входное видео и записывающие устройства, для повышения удобства дальнейшего использования приложения. В качестве конечного контрольного набора жестов было выбрано подмножество международной азбуки глухонемых. Причиной такого выбора была распространённость и свободное использование этого множества жестов определённым кругом людей.

Глава 2

Постановка задачи

Целью данной работы была разработка и реализация алгоритма, который получает на вход видеопоток, снятый с одной веб камеры, разбивает его на отдельные жесты, каждый из которых классифицирует в рамках выбранного подмножества из азбуки глухонемых.

Формат входного видео было решено максимально приблизить к естественным условиям использования.

Входная информация:

- В кадр может попадать весь человек, видео записывается в удобном формате, когда человек сидит перед видеокамерой и показывает жесты.
- В течении записи видео возможна смена освещения.
- Снимать можно как при дневном, так и при искусственном освещении.
- Время задержки на определённом жесте может быть минимальным, порядка пары секунд.
- На видео возможно шумовое фоновое или повторяющееся движение на заднем плане.

При разработке приложения основной упор делался на упрощение ввода информации в компьютер, поэтому одним из главных требований была минимизация ограничений на входное видео. В качестве записывающей аппаратуры было решено использовать только одну камеру, т.к. это существенно снижало как стоимость эксплуатации приложения, так и доступность его применения. Видео по формату должно быть таким, как если бы человек просто сел перед камерой, встроенной в компьютер или телефон и начал показывать жесты. Интервалы между жестами должны быть минимальны, что бы не задерживать поток вводимой информации. Так же было решено отказаться от дополнительных маркеров в пользу доступности приложения. Учитывая, что предполагалось записывать видео непосредственно перед камерой, использование цвета кожи как признака при локализации ладони было невозможно в связи с тем, что свет исходящий от монитора мог значительно исказить цветовую гамму на видео.

В качестве итогового множества жестов было выбрано подмножество латинской азбуки глухонемых, состоящее из 6-ти букв. Выбор был обусловлен естественностью и распространённостью этих жестов, а также большим потенциалом расширения набора только за счёт увеличения тренировочных данных, без дополнительной разработки.

Глава 3

Обзор существующих решений

Исследования по теме распознавания жестов проводились неоднократно, но, в связи с множеством вариаций формата входных данных и набором распознаваемых жестов, вопрос нельзя считать исчерпанным. Работы в данной области можно условно разделить на несколько групп.

1 Системы распознавания динамических жестов

В ряде работ ([14], [19]), за контрольное множество принимаются динамические жесты, состоящие из последовательных переходов руки из одного состояния в другое. Атомарным жестом в данном контексте являются движение руки вверх или вниз, приближение или отдаление. Распознавание, в таком случае, зачастую основано на скрытых Марковских цепях, что и является главной причиной не применимости исследований к статическим жестам, т.к. базой для распознавания является последовательность изменений, а не само состояние.

2 Использование дополнительных инструментов

В другой группе исследований [18] запись данных производится с использованием дополнительных устройств. Это может быть несколько

камер или же аппаратура производящая дополнительное излучение для получения карты глубины. По дополнительным данным строится трехмерное изображение по которому и происходит распознавание. Результаты, полученные в данной области, впечатляют, но вместе с этим, стоимость эксплуатирования приложений, написанных с использованием этих методов, увеличивается, а дополнительные условия на входные данные, снижают их мобильность.

3 *Ориентирование на цвет кожи*

В некоторых успешных работах [14] накладываются дополнительные требования на входные данные: освещение в котором производилась запись видео не должно искажать итоговую цветовую гамму. Это связано со способом поиска области изображения в котором находится ладонь, она выбирается по концентрации пикселей цвета кожи. В повседневной ситуации, например при записи видео со встроенной видеокамеры, такое условие зачастую нарушается.

4 *Использование маркеров*

Другим часто используемым требованием на входное видео является использование дополнительных маркеров, помогающих локализации ладони и построению её модели. Зачастую это перчатки, на которых каждый палец покрашен отдельным цветом [8].

Глава 4

Используемые методы

Проведённый анализ методов, используемых при распознавании жестов, позволил разделить алгоритм на этапы и выделить несколько подходов подходящих для достижения наилучших результатов.

4.1 Вычитание фона на изображении

Принцип, лежащий в основе алгоритмов вычитания фона, состоит в том, что определённый кадр или группа кадров выбирается за эталонный фон. Все последующие кадры сравниваются с ним и, если модуль разницы яркостей для пикселя превосходит порог, то пиксель считается передним планом, иначе — фоном. В работе используется адаптивный метод основанный на смеси нормальных распределений [10].

Фоновое значение для каждого пикселя представлено в виде набора троек : $[w, \mu, \delta^2]$ веса пикселя, математического ожидания и дисперсии, вес — представляет собой меру схожести между значением пикселя и фоном. В дальнейшем, будем называть такую тройку «процессом» пикселя. Такое представление позволяет алгоритму адаптироваться к зашумлённому фону. В текущей работе для каждого пикселя запоминается 5 его про-

цессов, значения которых он принимает с течением времени. На первом кадре видео происходит инициализация модели: $[1, c, \delta_{fix}^2]$, где c — яркость пикселя, δ_{fix}^2 — выбранное начальное значение.

Для каждого последующего кадра, для каждого пикселя производится ряд шагов:

1 Измеряется различие между поступившим пикселем и его фоновыми значениями: Если различие меньше порога, то процесс объявляется текущим. Если больше, то инициализируется новый процесс и, в зависимости от количества уже созданных процессов, либо добавляется в конец набора, либо заменяется процесс с наименьшим весом. Новый процесс помечается как текущий.

2 Значения текущего процесса обновляются:

$$\mu_t = (1 - \alpha_1) * \mu_{t-1} + \alpha_1 * c.$$

$$\delta_t^2 = (1 - \alpha_2) * \delta_{t-1}^2 + \alpha_2 * (c - \mu_t)^2.$$

$$w_t = (1 - \alpha_3) * w_{t-1} + \alpha_3.$$

Для всех остальных процессов пикселя обновляется только вес:

$$w_t = (1 - \alpha_3) * w_{t-1},$$

где $\alpha_1, \alpha_2, \alpha_3$ — выбранные параметры алгоритма, влияющие на скорость адаптации.

3 Пиксель классифицируется как фоновый, если вес превышает выбранный порог и как принадлежащий переднему плану в противном случае.

4.2 Гистограмма направленных градиентов

Гистограмма направленных градиентов [17] является дескриптором, широко используемым в распознавании образов. Основная идея состоит в том, что объект можно задать с помощью гистограмм ориентации градиента на изображении.

Алгоритм состоит из нескольких шагов:

1 *Вычисление градиента*

На первом этапе вычисления дескриптора стоит нормализация цвета и гамма-коррекция. Для чего производится свертка изображения с ядром, т.е. к каждому пикселю изображения применяется маска и его значение заменяется на сумму взвешенных значений пикселей окрестности.

2 *Группировка направлений*

На втором шаге изображение делится на ячейки, для каждой из которых производится подсчет гистограммы ориентаций градиентов. Форма ячеек может быть как прямоугольной, так и круглой формы, а весь диапазон направлений равномерно делится от 0 до 360 градусов, в случае, если вычисляется градиент без знака, или от 0 до 180, если градиент со знаком.

3 *Блоки дескрипторов*

Следующий шаг состоит в группировании маленьких ячеек в более крупные блоки, такое объединение требуется для нормализации значений ячеек. Зачастую после объединения ячейки могут входить одновременно в несколько разных блоков, и, как следствие, гистограмма будет нормализована несколько раз.

4 Нормализация блоков

На заключительном этапе производится нормализация блоков и объединение элементов гистограмм блоков для получения вектора дескриптора.

4.3 Машинное обучение

С помощью методов машинного обучения производится определение принадлежности жеста из входного потока к одной из выбранных групп.

4.3.1 Постановка задачи классификации

Для классификации объекта выделяется набор признаков, который считается достаточным для идентификации класса объекта, набор признаков представляется в виде n -мерного вектора. Классификатор может давать положительный, в случае принадлежности объекта к главному множеству, или отрицательный результат, в противном случае. Задача классификации является задачей с учителем, поэтому на вход подаётся тренировочное множество из m векторов: $x^{(1)} = [x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)}], x^{(2)}, \dots, x^{(m)}$. Каждому вектору ставится в соответствие ожидаемый результат классификации: $y^{(1)}, y^{(2)}, \dots, y^{(m)}$.

4.3.2 Логистическая регрессия

Метод логистической регрессии основан на представлении классификатора как параметрически заданной функции:

$$h_{\Theta}(x) = \frac{1}{1+e^{-\Theta^T x}}, \text{ где } \Theta = [\Theta_1, \Theta_2, \dots, \Theta_n].$$

С помощью тренировочного множества подбирается набор параметров

наиболее оптимальный для данной классификации. Определяется функция ошибки и с помощью метода градиентного спуска находится её минимум:

$$J(\Theta) = \frac{1}{2m} \sum_{i=1}^n (h_{\Theta}(x^{(i)}) - y^{(i)})^2 \rightarrow \min, \text{ где } y^i \in \{0, 1\}.$$

4.3.3 Support Vector Machines

Классификация в методе опорных векторов происходит с помощью разделения точек различных классов гиперплоскостью. Таких гиперплоскостей может быть много, поэтому в качестве меры качества выбранной является зазор между классами. Если существует гиперплоскость, разделяющая классы с максимальным зазором, то она называется оптимальной разделяющей гиперплоскостью, а соответствующий ей линейный классификатор называется оптимально разделяющим классификатором.

Строим разделяющую гиперплоскость, которая имеет вид (где w - перпендикуляр к разделяющей гиперплоскости):

$$w * x - b = 0.$$

Отдельно введём понятие ошибка классификации для данного элемента - ξ_i . Результат классификации, в таком случае, может принимать два значения: $y_i \in \{-1, 1\}$. Согласно данной классификации каждый вектор x должен удовлетворять такому условию:

$$y_i * (w * x_i - b) \geq 1 - \xi_i.$$

Для нахождения максимально возможного зазора между классами при

таким линейным разделением необходимо минимизировать следующую функцию:

$$\frac{1}{2} * ||w||^2 + C * \sum_{i=1}^n \xi_i \rightarrow \min.$$

4.3.4 Нелинейная SVM

Идея классификации с помощью нелинейной SVM практически полностью повторяет идеи линейной, с той лишь разницей, что каждое скалярное произведение заменяется нелинейной функцией ядра.

В данной работе в использовалось ядро Гаусса:

$$k(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\right).$$

4.3.5 Мультиклассификатор

Основой мультиклассификации был выбран алгоритм OneVsAll.

- 1 Необходимо произвести классификацию с m различными исходами.
- 2 Для каждого $i \in \{1 : m\}$ происходит классификация, где положительным исходом считается принадлежность данного объекта к i -ому классу и отрицательной в противном случае.
- 3 В результате получается m -мерный вектор вероятностей, принадлежности к каждому классу, на основе которого и происходит классификация.

Глава 5

Алгоритм

Входная информация - видео поток с одной вебкамеры. Видео может содержать разнородный фон и шумовые движения на заднем плане.

1 Предварительная обработка изображения

Каждый кадр (Рис.5.1) переводится в черно-белое изображение исходя из значения яркости для пикселей (Рис.5.2) и с помощью адаптивного алгоритма [10] вычитается фон (Рис.5.3). На полученной карте движения убирается шум и выделяются компоненты связности для каждой из которых считается вес.

2 Локализация ладони

Для каждой компоненты связности на кадре подсчитывается её вес до тех пор, пока значение не превысит пороговое. Считаем область наибольшей активности движения — ладонью. На последующих кадрах, в зависимости от того, является ли жест зафиксированным или происходит его смена, область ладони корректируется (Рис.5.4).

3 Разделение видео на жесты

В работе используется адаптивный метод вычитания фона и, как ре-

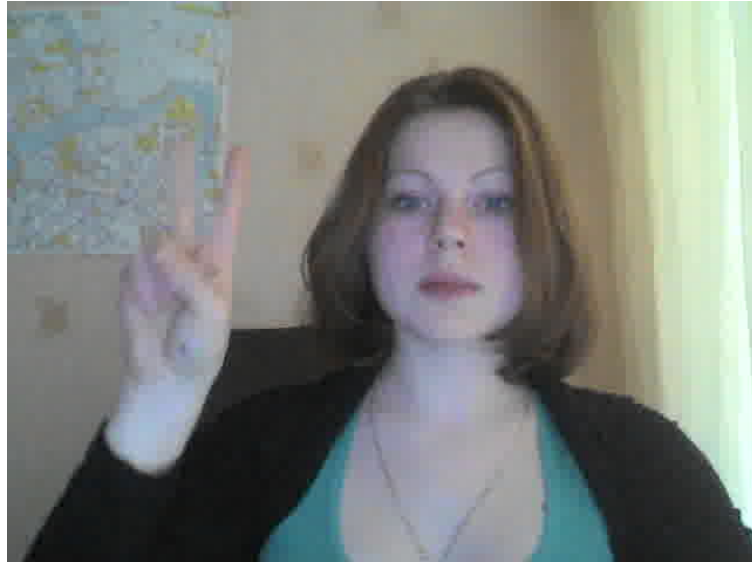


Рис. 5.1: Исходное изображение

зультат, резкие пики на графике постепенно затухают. Возрастание активности движения является началом смены жеста, а его затухание признак того, что жест устоялся и его можно классифицировать (Рис.5.5).

4 Получение признаков жеста

В качестве признаков для классификации было использовано два набора:

- *Моменты контура и габариты ладони*

На фрагменте изображения, где была локализована ладонь с помощью метода Канни[13] вычислялись контуры.

Моменты [6] - характеристика контура вычисляемая через суммирование всех его пикселей, вычисляется по формуле:

$$M_{ij} = \sum_{x=1}^{n_1} \sum_{y=1}^{n_2} x^i y^j I(x, y),$$

где $I(x, y)$ - бинарное значение изображения, равное 1, если контур проходит через пиксель с координатами (x, y) и 0 в противном слу-



Рис. 5.2: Значения яркостей изображения

чае, i и j - параметры момента.

Моменты вычисляемые по данной формуле зависят от выбора системы координат, поэтому в качестве признаков было решено использовать из нормализованную инвариантную модификацию, которая вычисляется в несколько этапов.

На первом этапе производится централизация относительно центра масс:

$$\mu_{ij} = \sum_{x=1}^{n_1} \sum_{y=1}^{n_2} (x - x_c)^i (y - y_c)^j I(x, y),$$

где $x_c = \frac{M_{10}}{M_{00}}$, $y_c = \frac{M_{01}}{M_{00}}$ - центры масс

Полученные результаты нормализуются:

$$\nu_{ij} = \frac{\mu_{ij}}{M_{00}^{\left(\frac{i+j}{2}+1\right)}}.$$

Моменты описывающие контур инвариантно вращению, отображения и масштаба:

$$\phi_0 = \nu_{20} + \nu_{02},$$

$$\phi_1 = (\nu_{20} - \nu_{02})^2 + 4\nu_{11}^2,$$



Рис. 5.3: Бинарное изображение, разделяющее исходное на передний и задний план

$$\phi_2 = (\nu_{30} - 3\nu_{12})^2 + (3 * \nu_{21} - \nu_{03})^2,$$

$$\phi_3 = (\nu_{30} + \nu_{12})^2 + (\nu_{21} + \nu_{03})^2,$$

$$\phi_4 = (\nu_{30} - 3 * \nu_{12}) * (\nu_{30} + \nu_{12})[(\nu_{30} + \nu_{12})^2 - 3 * (\nu_{21} + \nu_{03})^2] + (3 * \nu_{21} - \nu_{03})(\nu_{21} + \nu_{03})[3 * (\nu_{30} + \nu_{12})^2 - (\nu_{21} + \nu_{03})^2],$$

$$\phi_5 = (\nu_{20} - \nu_{02})[(\nu_{30} + \nu_{12})^2 - (\nu_{21} + \nu_{03})^2] + 4 * \nu_{11} * (\nu_{30} + \nu_{12})(\nu_{21} + \nu_{03}),$$

$$\phi_6 = (3 * \nu_{21} - \nu_{03})(\nu_{21} + \nu_{03})[3 * (\nu_{30} + \nu_{12})^2 - (\nu_{21} + \nu_{03})^2] - (\nu_{30} - 3 * \nu_{12})(\nu_{21} + \nu_{03})[3 * (\nu_{30} + \nu_{12})^2 - (\nu_{21} + \nu_{03})^2].$$

- *Гистограмма направленных градиентов*

Гистограмма направленных градиентов [15] является дескриптором особых точек, задающим объект через интенсивность изображения и направления границ объектов.

5 Классификация жеста

Каждый кадр во временном промежутке, когда жест считается устоявшимся, классифицируется с помощью метода опорных векторов [3], из всех полученных результатов классификации выбирается тот, который встретился более, чем в половине случаев, иначе жест считается



Рис. 5.4: Изменение активности движения на видео
не принадлежащим контрольному множеству.

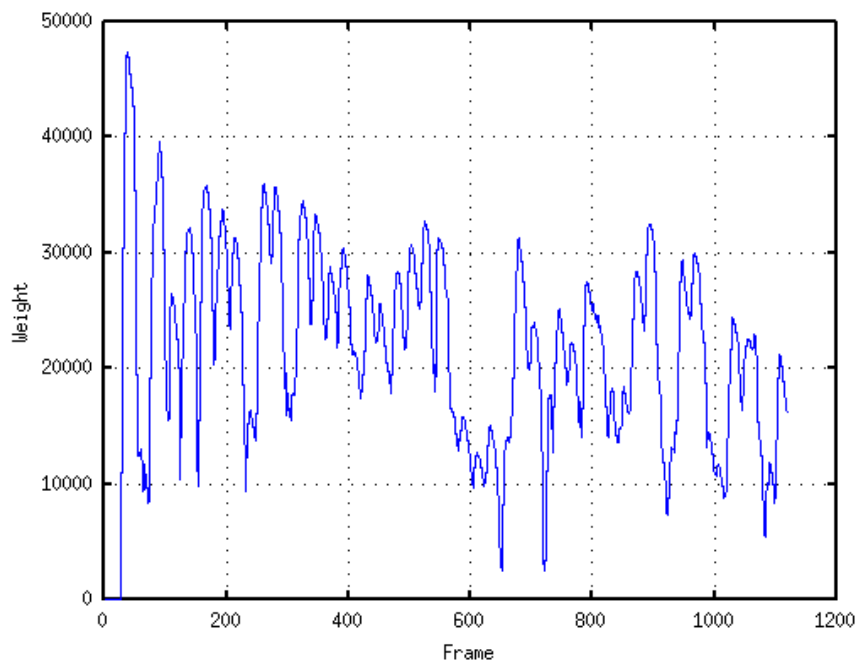


Рис. 5.5: Изменение активности движения на видео

Глава 6

Реализация

6.1 Проектирование

В качестве языка разработки приложения был выбран язык Octave, который, благодаря своей «заточенности» на программирование вычислений и оптимизированной работе с матрицами, позволил сосредоточиться на алгоритмической составляющей приложения. Так же на выбор оказало влияние наличие встроенных библиотек компьютерного зрения и схожесть языка Octave с более распространённым, но коммерческим языком Matlab.

Входные данные записывались с помощью обычной веб камеры, в ситуации, когда человек сидит непосредственно перед ней. Важным требованием при записи информации, было доступность способа, и использование встроенной в компьютер камеры полностью отвечало этому требованию. Тем не менее, запись видео в непосредственной близости от возможного источника света наложило определённые требования на дальнейшую разработку: пришлось отказаться от одного из наиболее очевидных признаков нахождения ладони - цвета кожи, так как свет от монитора компьютера может привести к сильному искажению цветов на полученном видео.

В качестве контрольного набора данных было выбрано подмножество языка глухонемых: А, С, V, G, H, Y

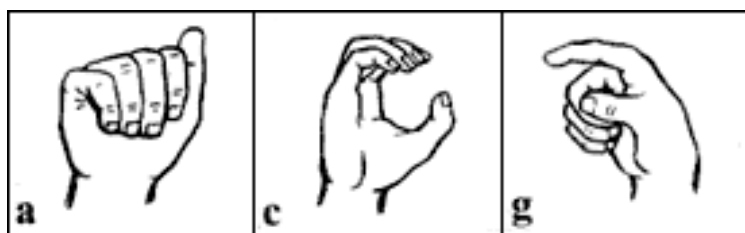


Рис. 6.1: Буквы: А, С, G

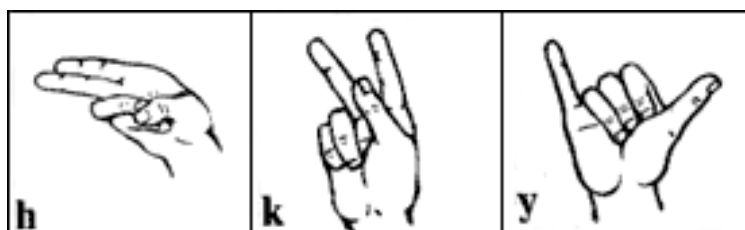


Рис. 6.2: Буквы: H, K, Y

Выбор такого множества символов был обусловлен естественностью и распространённостью жестов среди определённой группы людей и возможностью расширения контрольного набора жестов только за счёт увеличения тренировочных данных.

6.2 Особенности реализации

Работа с большим объёмом данных

Использовалось видео с расширением 480x640, которое разделялось на изображения из расчёта 24 кадра в секунду. В результате видео, длящееся всего одну минуту, представлялось в виде более тысячи изображений, каждое из которых является совокупностью из $480 * 640 = 307200$ пикселей. Многие алгоритмы компьютерного зрения, как, например, используемый в данной работе алгоритм вычитания фона, требуют обработки каждого

пикселя изображения. Как следствие, без аппаратной реализации время работы алгоритма резко возрастает, что существенно сказывается на процессе разработки.

Внешние условия

Большинство алгоритмов компьютерного зрения очень чувствительны к внешним условиям. Небольшие изменения освещения могут значительно исказить их работу, сделав методы неэффективными или вовсе неприменимыми. Именно поэтому задача о постановке ограничений на входное видео имело первостепенное значение, необходимо было найти баланс между удобством использования приложения и эффективностью его работы.

Эвристические методы

Как алгоритмы искусственного интеллекта в обучении, так и алгоритмы компьютерного зрения зависят от многих эвристических параметров. Например, скорость адаптации алгоритма при детектировании движения на видео, или пороговые значения для классификатора. Настройка таких параметров - это продолжительный процесс, требующий учёта всех возможных исходных данных для получения лучшего результата в среднем случае.

Сбор данных и обучение системы

Процесс обучения системы можно разделить на несколько последовательных этапов: сбор данных, их разметка, обучение параметров классификатора.

- На первом этапе необходимо получить большое количество данных при различных вариациях внешних условий. Это должно быть видео с разными людьми, с различным фоном и при разном освещении.
- Второй этап прост в исполнении, но достаточно трудоёмок, т.к. для каждого классифицируемого изображения необходимо указать желаемый результат классификации.
- Последний этап представляет собой само обучение классификатора, т.е. минимизации функции погрешности для размеченных данных.

6.3 Тестирование

Тестирование разработанного приложения производилось на нескольких видео, записанных при разном освещении и с различным фоном. От видео к видео менялась скорость смены жеста для того, что бы проверить эффективность этапа разделения на жесты. На всех видео присутствует один и тот же человек под которого производилось обучение системы классификаторов. В общей сумме количество жестов составило около двухсот.

Глава 7

Результат работы

В ходе работы был реализован разработанный алгоритм на языке Octave.

Обучение классификатора производилось на множестве из 75 кадров для каждого жеста. Настройка классификатора производилась под конкретного человека, для получения лучших результатов классификации.

Основное требование - удобство использования получилось соблюсти. Информация поступает с одной камеры, которая может быть встроенной в компьютер, что делает возможным использовать приложения «с нуля», не производя долгую настройку и калибровку аппаратуры. Ставка на цвет кожи не производилась, что позволяет использовать приложения в помещении со специфическим освещением, искажающим цвета объектов.

Для оценки классификации использовались стандартные метрики:

$$precision = \frac{tp}{tp+fp},$$

$$recall = \frac{tp}{tp+fn}.$$

tp(true positive) - количество объектов на которые был дан положительный

ответ в том случае, когда он и требовался;

fp(false positive) - положительный ответ, когда был верен отрицательный;

fn(false negative) - отрицательный ответ, когда требовался положительный.

В работе под «false positive» понимаются те случаи, когда классификатору на вход поступал жест не относящийся к контрольной группе, но который, тем не менее, был классифицирован. Под «false negative», в свою очередь, понимается количество жестов ложно классифицированных как не относящие к контрольной группе.

	Моменты контура	Гистограмма направленных градиентов
Precision	0.90643	0.96685
Recall	0.91716	0.97765
Percent	0.82703	0.93514

Таблица 7.1: Значения метрик в зависимости от выбранных признаков классификатора

Глава 8

Заключение

В ходе дипломной работы были изучены существующие подходы к распознаванию образов, на основе которых был разработан и реализован алгоритм по распознаванию латинской азбуки жестов на видеопотоке. Ряд экспериментов на зависимых от человека данных показал эффективность полученного метода.

8.1 Дальнейшее развитие

Дальнейшую работу над проблемой можно произвести в нескольких направлениях:

1 *Увеличение скорости работы приложения*

Требование относительно работы приложения в режиме реального времени выполнить не удалось, распознавание происходит с задержкой и, как следствие, возможно только на записи видео. Наиболее вычислительно сложным является этап разделения изображения на передний и задний план. Предполагается, что аппаратная реализация этой части алгоритма даст большой прирост в скорости и появится возможность делать вычисления "на лету".

2 Расширение числа распознаваемых жестов

На текущем этапе приложение в качестве итогового набора распознавания было выбрано подмножество латинского алфавита жестов. Его расширение вплоть до всего алфавита не представляется сложной задачей и может быть осуществлено только за счёт добавление новых данных для обучения.

3 Обучение не зависимо от человека

В данной работе обучение и распознавание жестов производилось на одном и том же человеке. В дальнейшем одним из наиболее приоритетных направлений является проверка работоспособности разработанного метода при распознавании жестов в не зависимости от человека.

Литература

- [1] В.Г.Абакумов, Е.Ю.Ломакина «Автоматическое распознавание жестов в интеллектуальных системах» // «Электротехнические и компьютерные системы», 2010.
- [2] В.Н. Вапник, А.Я. Червоненкис «Теория распознавания образов», издательство: «Наука», 1974.
- [3] «Введение В Контурный Анализ», 2003 / под ред. Я.А. Фурмана.
- [4] А.Л.Воскресенский, С.Н.Ильин, М.Zeleznu «О распознавании жестов языка глухих», 2010.
- [5] Д.Форсайт, Ж.Понс «Компьютерное зрение», издательство: «Вильямс», 2004.
- [6] Л.Шапиро, Дж.Стокман «Компьютерное зрение», издательство: «Бинном. Лаборатория знаний», 2006.
- [7] «Язык жестов» :- URL: <http://jestov.net/>
- [8] Aran, Oya «Vision based sign language recognition: modeling and recognizing isolated signs with manual and non-manual components», 2008.
- [9] A.Barr, Edvard A. Feigenbaum «The Handbook of Artifical Intelligence», издательство: «Slategray Ventures», 1990.

- [10] Chris Stauffer, W. Eric L. Grimson «Learning Patterns of Activity Using Real-Time Tracking», 2000.
- [11] Gary Bradski, Adrian Kaehler «Learning OpenCV», издательство: «O'RELLY», 2008.
- [12] Hsu, Chih-Wei, Chang, Chih-Chung; and Lin, Chih-Jen «A Practical Guide to Support Vector Classification», 2003.
- [13] John Canny «A Computational Approach to Edge Detection», 1986.
- [14] Jorg Zieren, Karl-friedrich Kraiss «Robust person-independent visual sign language recognition», 2005.
- [15] Karl Pearson «On Lines and Planes of Closest Fit to System of Points in Space», 1901.
- [16] Lemeshow, David W. Hosmer «Applied logistic regression», 2000.
- [17] Navneet Dalal, Bill Triggs «Histograms of Oriented Gradients for Human Detection», 2005.
- [18] Simon Lang «Sign Language Recognition with Kinect», 2011.
- [19] Thad Starner, Joshua Weaver, Alex Pentland «Real-time american sign language recognition using desk and wearable computer based video», 1998.
- [20] Tom M. Mitchell «Machine Learning», издательство: «Cambridge University Press», 1997.