Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Южно-Уральский государственный университет»

(национальный исследовательский университет)

Факультет «Математики, механики и компьютерных наук»

Кафедра «Вычислительная механика сплошных сред»

РАБОТА ПРОВЕРЕНА Допустить к Защите

Рецензент, Заведующий кафедрой,

к.ф.-м.н., доцент кафедры МиФА д. ф.-м. н., профессор

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/М.А. Корытова \_\_\_\_\_\_\_\_\_\_\_\_\_\_/Ковалев Ю.М.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2016 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2016 г.

**Детектирование ключевых точек на лице человека с использованием сверточных нейронных сетей**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ЮУрГУ – 010800.62.2015.838.ВКР

Руководитель

к., ф.-м. н., доцент кафедры

ВМСС

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Н.Л. Клиначева

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2016 г.

Автор работы

студент группы ММиКН-460

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/А.О. Литвинов

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2016 г.

Нормоконтролер – к.ф.-м.н.,

доцент кафедры ВМСС

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/В.К.Рябинин

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2016 г.

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Южно-Уральский государственный университет»

(национальный исследовательский университет)

Факультет «Математики, механики и компьютерных наук»

Кафедра «Вычислительная механика сплошных сред»

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ Заведующий кафедрой,  д. ф.-м. н., профессор  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Ю.М. Ковалев  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2016 г. |

**ЗАДАНИЕ**  
студенту группы ММиКН-473

Литвинову Андрею Олеговичу

на выполнение выпускной квалификационной работы

по направлению 011000.62.–Механика. Прикладная математика.

1. **Тема выпускной квалификационной работы** Детектирование ключевых точек на лице человека с использованием сверточных нейронных сетей

**(Утверждена приказом ректора от 27.04.2015 № 838)**

1. **Перечень подлежащих исследованию вопросов**

2.1 Обзор литературы по решению задачи о распространении плоской ударной волны в воздухе, взаимодействии с гетерогенным слоем

2.2 Составление программы расчёта.

2.3 Решение задачи для заданных начальных условий.

1. **Календарный план подготовки выпускной квалификационной работы**

|  |  |  |
| --- | --- | --- |
| Наименование этапов выпускной квалификационной работы | Срок выполнения этапов работы | Отметка о выполнении |
| 1.Обзор литературы | 10.02.16-03.03.16 | выполнено |
| 2. Изучение численного метода для решения поставленной задачи. Решение тестовой задачи. | 04.03.16-25.04.16 | выполнено |
| 3.Получение результатов, формулировка выводов, структурирование текста | 25.04.16-15.05.16 | выполнено |
| 4.Подготовка текста | 16.05.15-01.06.15 | выполнено |
| 5. Проверка и рецензирование работы руководителем, исправление замечаний. Подготовка доклада и текста выступления. Внешнее рецензирование. | 02.06.15-13.06.15 | выполнено |
| 6. Защита выпускной квалификационной работы. | 22.06.16 |  |

**4. Дата выдачи задания 10 февраля 2016 г.**

**Руководитель работы**

к.ф.-м.н., доцент кафедры ВМСС \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Н.Л. Клиначева

Задание принял к исполнению**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/**А.О. Литвинов

УДК 533.6.011.1

**Литвинов А.О.**

Построение математической модели многокомпонентной смеси с учетом сжимаемости частиц инвариантной к преобразованиям Галилея. Литвинов А.О. – Челябинск, 2016. – 35с.

В дипломной работе реализован модифицированный метод крупных частиц для расчета течений в сложных задачах газодинамики. Исследовано влияние частиц на параметры УВ, падающей на гетерогенный слой. Проведен анализ математической модели, предложенной П.Б. Вайнштейном, на инвариантность относительно преобразований Галилея.

В приложении приведен текст программы на языке python

Список лит. – 20 назв.

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 6](#_Toc451593018)

[ПОСТАНОВКА ЗАДАЧИ 7](#_Toc451593019)

[АНАЛОГИЧНЫЕ МЕТОДЫ РЕШЕНИЯ 8](#_Toc451593020)

[СТРУКТУРА СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ 9](#_Toc451593021)

[ДЕТЕКТИРОВАНИЕ ЧЕЛОВЕЧЕСКОГО ЛИЦА НА ИЗОБРАЖЕНИИ ДЛЯ ВХОДА СВЕРТОЧНОЙ СЕТИ 11](#_Toc451593022)

[НАХОЖДЕНИЕ ТОЧЕК ЛИЦА ПО ДЕТЕКТИРОВАННОМУ СЕКТОРУ 16](#_Toc451593023)

[ПРИНЦИП РАБОТЫ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ 16](#_Toc451593024)

[ПРИНЦИП ОБУЧЕНИЯ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ 18](#_Toc451593025)

[МЕТОДЫ ПРИМЕНЯЕМЫЕ ПРИ ОБУЧЕНИИ 21](#_Toc451593026)

[РЕЗУЛЬТАТЫ 25](#_Toc451593027)

[ЗАКЛЮЧЕНИЕ 26](#_Toc451593028)

[ЛИТЕРАТУРА 27](#_Toc451593029)

[ПРИЛОЖЕНИЕ 28](#_Toc451593030)

# ВВЕДЕНИЕ

Искусственные нейронные сети в настоящее время получили широкое распространение в различных сферах человеческой деятельности, таких как распознавание и классификация образов, прогнозирование, управление. Пример конкретных задач – автономная машина, распознавание изображений, речи, текстовые чат боты. Это произошло благодаря тому, что они имеют возможность синтезировать какие-либо алгоритмы, вычленяя из больших выборок данных произвольные закономерности. Изначально они построены по примеру биологических нейронных сетей – сетей нервных клеток животного. Один из видов искусственных нейронных сетей - сверточные нейронные сети, в основном применяющиеся для работы с изображениями. Причину этого можно увидеть в предпосылках их зарождения. Сверточные сети построены по примеру обработки визуальной информации у живых существ – в мозге есть области, выделяющие какие-либо инвариантные закономерности в изображениях – например линии.

Тип задачи, решаемой в данной работе, относиться к классу регрессионных задач, то есть восстановление каких-либо зависимостей. Аналогичные методы решения возможно применять к схожим проблемам – распознавание рукописного текста, номеров автомобилей, дорожных знаков.

# ПОСТАНОВКА ЗАДАЧИ

Данные получаемые с видеокамеры(кадр) в момент времени обозначим . Каждый кадр есть матрица размерности , каждый элемент принадлежит множеству , где – ширина кадра, - высота кадра. Видеопоследовательность есть упорядоченная последовательность кадров (), где – количество кадров в видеопоследовательности.

Сформулируем задачу детектирования точек на лице человека. По имеющемуся кадру предсказать положение точек лица человека, предполагая при этом, что в кадре находится только один человека, на расстоянии от 0.5 до 2 метров от камеры. Требуется предсказать положение 8 точек:

* Центр подбородка
* Левый край губ
* Правый край губ
* Центр носа
* Левую височную область
* Левый край левого глаза
* Правую височную область
* Правый край правого глаза

В данной работе необходимо из видеопоследовательности в режиме реального времени выделять точки лица. В качестве исходных данных выступает видеопоследовательность, полученная с неподвижной камеры, с разрешением пикселей в формате RGB глубиной 8 бит на один канал и частотой кадров 30 в секунду.

Заданные входные данные накладывают на возможные методы решения ограничения в плане оптимизации

# АНАЛЬТЕРНАТИВНЫЕ МЕТОДЫ РЕШЕНИЯ

**Метод главных компонент**

Метод главных компонент (Principal Component Analysis, {PCA}) применяется для сжатия информации без существенных потерь информативности. Он состоит в линейном ортогональном преобразовании входного вектора xx размерности NN в выходной вектор yy размерности MM, N>MN>M. При этом компоненты вектора yy являются некоррелированными и, следовательно, общая дисперсия после преобразования остается неизменной. Матрица XX состоит из всех примеров изображений обучающего набора. Решив уравнение Λ=ΦTΣΦΛ=ΦTΣΦ, получаем матрицу собственных векторов ΦΦ, где ΣΣ - ковариационная матрица дляxx, а ΛΛ - диагональная матрица собственных чисел. Выбрав из ΦΦ подматрицу ΦMΦM, соответствующую MM наибольшим собственным числам, получим, что преобразование y=ΨTMx˜y=ΨMTx~, где x˜=x−x¯¯¯x~=x−x¯ - нормализованный вектор с нулевым математическим ожиданием, характеризует большую часть общей дисперсии и отражает наиболее существенные изменения xx. Выбор первых MM главных компонент разбивает векторное пространство на главное (собственное) пространство F={Φi}Mi=1F={Φi}i=1M, содержащее главные компоненты, и его ортогональное дополнение F={Φi}Ni=M+1F={Φi}i=M+1N. В качестве индикаторов принадлежности в методе главных компонент используют:

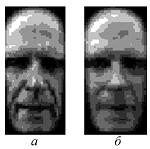
DIFS - distance in feature space, расстояние от образа анализируемого изображения в собственном пространстве, до эталонного образа;

DFFS - distance from feature space, расстояние от представления анализируемого изображения в пространстве наблюдения до проекции эталона в собственном пространстве.

[](http://wiki.technicalvision.ru/index.php/%D0%A4%D0%B0%D0%B9%D0%BB:7-6-1.jpg)

Пример изображений собственных векторов (собственные лица)

Применение для задачи распознавания человека по изображению лица имеет следующий вид. Входные векторы представляют собой отцентрированные и приведенные к единому масштабу изображения лиц. Собственные векторы, вычисленные для всего набора изображений лиц, называются собственными лицами (eigenfaces). Метод главных компонент в применении к изображениям лиц также называют методом собственных лиц (рис. 1). С помощью вычисленных ранее матриц входное изображение разлагается на набор линейных коэффициентов, называемых главными компонентами. Сумма NN первых главных компонент, умноженных на соответствующие собственные векторы, является аппроксимацией изображения порядка NN (рис. 2).

[](http://wiki.technicalvision.ru/index.php/%D0%A4%D0%B0%D0%B9%D0%BB:7-6-2.jpg)

Нормализованное изображение лица (*а*а) и его реконструкция по 8585 главным компонентам (*б*б)

Для каждого изображения лица вычисляются его главные компоненты. Обычно берется от 55 до 200200 главных компонент. Остальные компоненты кодируют мелкие различия между лицами и шум. Процесс распознавания заключается в сравнении главных компонент неизвестного изображения с компонентами всех остальных изображений. Для этого обычно применяют какую-либо метрику (простейший случай - Евклидово расстояние). Дополнительное повышение надежности достигается за счет дополнительного применения анализа главных компонент к отдельным участкам лица таким, как глаза, нос, рот.

Также метод главных компонент применяется для обнаружения лица на изображении. Для лиц значения компонент в собственном пространстве имеют большие значения, а в дополнении собственного пространства - близки к нулю. По этому факту можно обнаружить, является ли входное изображение лицом. Для этого проверяется величина ошибки реконструкции; чем больше ошибка, тем больше вероятность, что это не лицо. При наличии в наборе изображений лиц вариаций таких, как раса, пол, эмоции, освещение, будут появляться компоненты, величина которых в основном определяется этими факторами. Поэтому по значениям соответствующих главных компонент можно определить, например, расу или пол человека. Основные недостатки PCA таковы. Метод собственных лиц требует для своего применения идеализированных условий таких, как единые параметры освещенности, нейтральное выражение лица, отсутствие помех вроде очков и бород. При несоблюдении этих условий главные компоненты не будут отражать межклассовые вариации. Например, при различных условиях освещенности метод собственных лиц практически неприменим, поскольку первые главные компоненты преимущественно отражают изменения освещения, и сравнение выдает изображения, имеющие похожий уровень освещенности.

Вычисление набора собственных векторов отличается высокой трудоемкостью. Один из способов - это свертка изображений по строкам и столбцам; в такой форме представление изображения имеет на порядок меньший размер, вычисления и распознавание происходит быстрее, но восстановить исходное изображение уже невозможно.

**Линейный дискриминантный анализ**

Линейный дискриминантный анализ (линейный дискриминант Фишера Linear Discriminant Analysis, {LDA}) использует такую проекцию пространства изображений на пространство признаков, которая минимизирует внутриклассовое и максимизирует межклассовое расстояние в пространстве признаков. В этих методах предполагается, что классы линейно разделимы.

Матрица WW для проецирования пространства изображения на пространство признаков выбирается из следующего условия:

Wopt=argmaxWWTSBWWTSWW,Wopt=argmaxWWTSBWWTSWW,

где SBSB - матрица межклассовой дисперсии, SWSW - матрица внутриклассовой дисперсии.

Может существовать до c−1c−1 векторов составляющих базис пространства признаков, где cc - общее число классов. С помощью этих векторов пространство изображений переводится в пространство признаков.

Поскольку работа непосредственно с матрицей SW∈Rn×nSW∈Rn×n затруднительна из-за ее размерности, используется предварительное уменьшение размерности с помощью метода главных компонент, и затем вычисления производятся в пространстве меньшей размерности:

Wfld=argmaxWWTWTpcaSBWpcaWWTWTpcaSWWpcaW,Wfld=argmaxWWTWpcaTSBWpcaWWTWpcaTSWWpcaW,

где WpcaWpca - матрица для проецирования в пространство меньшей размерности (пространство главных компонент).

Обычно тренировочный набор содержит изображения лиц при нескольких базовых условиях освещенности, на основе которых при помощи линейных комбинаций можно получить любые другие условия освещенности. Этот метод дает высокую точность распознавания (около 96{\%}) для широкого диапазона условий освещенности, различных выражений лица и наличия или отсутствия очков. Однако остаются невыясненными вопросы, применим ли этот метод для поиска в больших базах данных, может ли метод работать, когда в тренировочной выборке для некоторых лиц имеется изображение только в одних условиях освещенности. Для задачи детектирования лица с помощью LDA-классы лиц и "не лиц" разбивают на подклассы. Вышеописанный метод основывается на предположении о линейной разделимости классов в пространстве изображений. В общем случае такое предположение несправедливо. Инструмент для построения сложных разделяющих поверхностей предлагают нейросетевые методы.

**Синтез объектов линейных классов**

Данный метод позволяет синтезировать новые изображения объекта (и в частности, изображения лица) для разных ракурсов. Имеется тренировочный набор изображений лиц и только одно изображение нового объекта в определенном ракурсе. Тренировочный набор состоит из изображений объектов того же класса (класс лиц в данном случае), что и новый объект, и включает в себя изображения различных лиц, причем для каждого лица имеются его изображения в широком диапазоне ракурсов. Для нового объекта, имеющего изображение XAXA в ракурсе AA, осуществляется линейное разложение на изображения объектов из тренировочного набора в том же ракурсе, с вычислением коэффициентов αi:XA=∑qi=1αiXAi,αi:XA=∑i=1qαiXiA, где qq - количество объектов в тренировочном наборе. Синтез изображения XBXB в новом ракурсе BB для нового объекта осуществляется сложением изображений из тренировочного набора в ракурсе BB с теми же коэффициентами: XB=∑qi=1αiXBiXB=∑i=1qαiXiB. Таким образом, метод позволяет синтезировать изображения нового объекта в различных ракурсах по изображению в одном ракурсе без привлечения сложных трехмерных моделей. Данный метод является перспективным для синтеза изображений в новых ракурсах без привлечения сложных трехмерных моделей, однако вопрос о качестве и количестве примеров в тренировочном наборе остается открытым.

**Гибкие контурные модели**

В данных методах распознавание производится на основе сравнения контуров лица. Контуры обычно извлекаются для линий головы, ушей, губ, носа, бровей и глаз. Контуры представлены ключевыми позициями, между которыми положение точек, принадлежащих контуру, вычисляются интерполированием. Для локализации контуров в различных методах используется как априорная информация, так и информация, полученная в результате анализа тренировочного набора. Обычно ключевые точки размещаются вручную на наборе тренировочных изображений. При поиске контуров нового лица используется метод симуляции отжига с целевой функцией из двух составляющих. Для первой из них ищется максимум при соответствии интенсивностей пикселов, извлеченных на перпендикулярной контуру линии, аналогичным пикселам из тренировочной выборки, для второй - при совпадении контура с формой контуров тренировочных примеров. Таким образом, извлекается контур черт лица. Для сравнения изображений используются значения главных компонент, вычисленные на наборе векторов, представляющих собой координаты ключевых точек. Главной задачей при распознавании по контурам является правильное выделение этих контуров. В общем виде эта задача по сложности сравнима непосредственно с распознаванием изображений.

**Сравнение эластичных графов**

В этом методе (Elastic Bunch Graph Matching) лицо представляется в виде графа, вершины которого расположены на ключевых точках лица таких, как контуры головы, губ, носы, и их крайних точках. Каждая грань помечена расстояниями между ее вершинами. В каждой такой точке вычисляются коэффициенты разложения по функциям Габора для пяти различных частот и восьми ориентаций. Набор таких коэффициентов J={Jj}J={Jj} называется *джетом*джетом (jet). Джеты характеризуют локальные области изображений и служат для двух целей: во-первых, для нахождения точек соответствия в заданной области на двух различных изображениях; во-вторых - для сравнения двух соответствующих областей различных изображений. Каждый коэффициент Jj=ajexp(iϕj)Jj=ajexp⁡(iϕj) для точек из одной области различных изображений характеризуется амплитудой ajaj, которая медленно меняется с изменением положения точки, и фазой ϕjϕj, которая вращается со скоростью, пропорциональной частоте волнового вектора базисной функции. Поэтому в простейшем случае для поиска на новом изображении точки с аналогичными характеристиками в функции подобия фазу не учитывают:

Sa(J,J′)=∑jaja′j∑ja2j∑ja′2j−−−−−−−−−−√.Sa(J,J′)=∑jajaj′∑jaj2∑jaj′2.

Функция подобия с одним джетом в фиксированной позиции и другим с переменной позицией является достаточно гладкой, для того чтобы получить быструю и надежную сходимость при поиске с применением простейших методов таких, как диффузия или градиентный спуск. Более совершенные функции подобия привлекают информацию о фазе. Для различных ракурсов соответствующие ключевые точки отмечаются вручную на тренировочном наборе. Кроме того, чтобы для одного и того же лица представить различные вариации его изображения в одном и том же графе, для каждой точки используются несколько джетов, каждый из которых может соответствовать различным локальным характеристикам данной точки, например, открытому и закрытому глазу. Процесс распознавания неизвестного лица состоит в сравнении графа изображения лица GIGI со всеми остальными графами из набора BB при помощи функции подобия

SB(GI,B)=1N∑nmaxmSϕ(JIn,JBmn)−λE∑e(ΔxIe−ΔxBe)2(ΔxBe)2.SB(GI,B)=1N∑nmaxmSϕ(JnI,JnBm)−λE∑e(ΔxeI−ΔxeB)2(ΔxeB)2.

Левая сумма характеризует подобие джетов, вычисленное с применением фазочувствительной функции, правая - топографическое соответствие, которое пропорционально квадрату разности расстояний между соответствующими вершинами сравниваемых изображений, NN - количество вершин, EE - количество граней, λλ - коэффициент относительной важности топографической информации.

В представленном выше виде метод способен достаточно надежно распознавать при изменениях ракурса до 2020° ; при больших углах точность распознавания резко уменьшается, функция подобия оказывается более чувствительной к ракурсу, чем к межклассовым различиям. Дальнейшее развитие метода заключается в извлечении коэффициентов важности на основе анализа обучающей выборки. Для каждого джета симплекс-методом вычисляется коэффициент важности, который затем используется в функции подобия. Коэффициенты важности вычисляются из условия максимизации функции подобия для одного и того же лица и минимизации - для различных лиц. Существуют также более ранние разновидности этого метода, которые не используют изначально определенные ключевые точки и структуры графа. Одни из них используют для сравнения решетки джетов, наложенные на изображение, рис. 3. В неизвестном изображении отыскиваются точки соответствия, и затем по найденным точкам строится искаженная решетка и измеряется мера ее искажения для определения наиболее похожего изображения. В других методах точки извлечения джетов изначально образуют решетку, а затем наименее пригодные для распознавания точки отсеиваются в процессе обучения.

[](http://wiki.technicalvision.ru/index.php/%D0%A4%D0%B0%D0%B9%D0%BB:7-6-3.jpg)

Наложенная на изображение эластичная решетка и ее искаженная версия

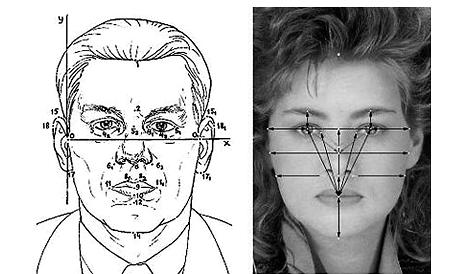
**Методы, основанные на геометрических характеристиках лица**

Один из самых первых методов - это анализ геометрических характеристик лица. Изначально он применялся в криминалистике и был там детально разработан. Потом появились компьютерные реализации этого метода. Суть его заключается в выделении набора ключевых точек (или областей) лица и последующем выделении набора признаков. Каждый признак является либо расстоянием между ключевыми точками, либо отношением таких расстояний. В отличие от метода сравнения эластичных графов, здесь расстояния выбираются не как дуги графов. Наборы наиболее информативных признаков выделяются экспериментально.

Ключевыми точками могут быть уголки глаз, губ, кончик носа, центр глаза и т. п. рис. 4. В качестве ключевых областей могут служить прямоугольные области, включающие в себя: глаза, нос, рот.

В процессе распознавания сравниваются признаки неизвестного лица с признаками, хранящимися в базе. Задача нахождения ключевых точек приближается к трудоемкости непосредственно распознавания, и правильное нахождение ключевых точек на изображении во многом определяет успех распознавания. Поэтому изображение лица человека должно быть без помех, мешающих процессу поиска ключевых точек. К таким помехам относят очки, бороды, украшения, элементы прически и макияжа. Освещение желательно равномерное и одинаковое для всех изображений. Кроме того, изображение лица должно иметь фронтальный ракурс, возможно, с небольшими отклонениями. Выражение лица должно быть нейтральным. Это связано с тем, что в большинстве методов нет модели учета таких изменений.

Таким образом, данный метод предъявляет достаточно строгие требования к условиям съемки и нуждается в надежном механизме нахождения ключевых точек для общего случая. Кроме того, требуется применение более совершенных методов классификации или построения модели изменений. В общем случае этот метод не является самым оптимальным, однако для некоторых специфических задач все же перспективен. К таким задачам можно отнести документный контроль, когда требуется сравнить изображение лица, полученного в текущий момент, с фотографией в документе. При этом других изображений этого человека не имеется, и, следовательно, механизмы классификации, основанные на анализе тренировочного набора, недоступны.

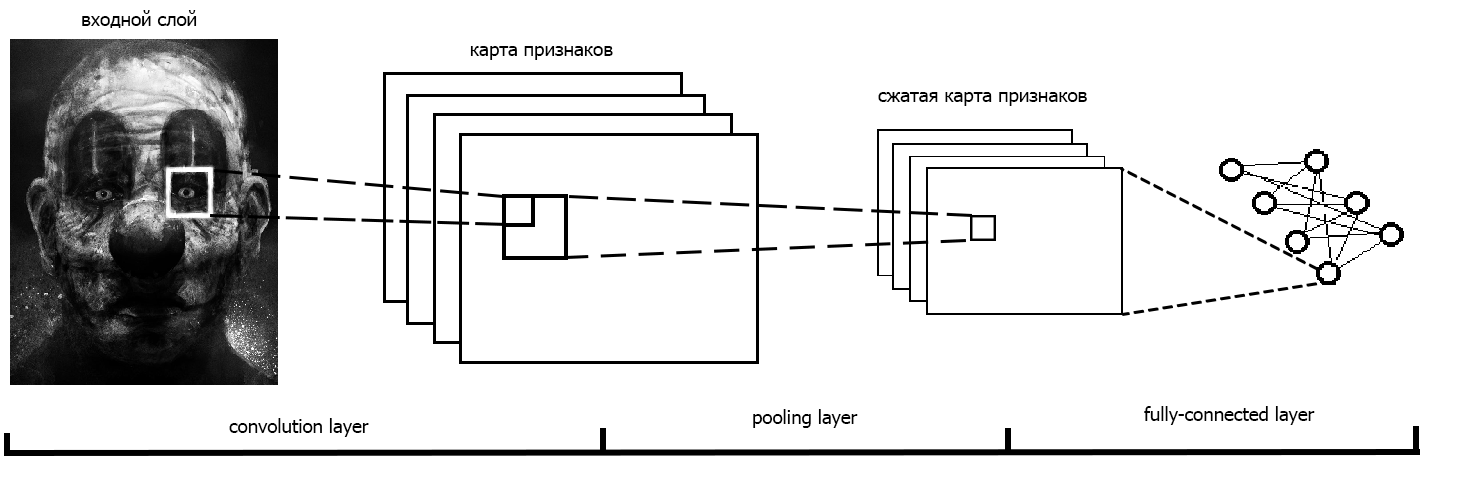
[](http://wiki.technicalvision.ru/index.php/%D0%A4%D0%B0%D0%B9%D0%BB:7-6-4.jpg)

Идентификационные точки и расстояния: *а*а - используемые при криминалистической экспертизе; *б*б - наиболее часто применяемые при построении автоматизированных систем идентификации

# СТРУКТУРА СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

Сверточные нейронные сети содержат три типа слоев:

1. **Convolutional(сверточные).** В операции свертки используется лишь ограниченная матрица весов небольшого размера, которую «двигают» по всему обрабатываемому слою (в самом начале по входным данным), формируя после каждого сдвига сигнал активации для нейрона следующего слоя с аналогичной позицией. То есть для различных нейронов выходного слоя используются общие веса — матрица весов, которую также называют набором весов или ядром свертки. Она построена таким образом, что графически кодирует какой-либо один признак, например, наличие наклонной линии под определенным углом. Тогда следующий слой, получившийся в результате операции свертки такой матрицей весов, показывает наличие данной наклонной линии в обрабатываемом слое и ее координаты, формируя так называемую карту признаков (feature map). В сверточной нейронной сети количество ядер, как правило много: они кодируют признаки исходных данных.
2. **Pooling, subsampling(операция субдискретизации).** После каждого сверточного слоя, может быть pooling слой. Pooling слой берет небольшие прямоугольные блоки из сверточного слоя и вычисляет один выход из этого блока. Есть несколько способов сделать эту операцию, например, взяв среднее, максимальное или какую-либо линейную комбинацию нейронов в блоке.
3. **Fully-Connected(полносвязная).** После нескольких convolutional и max-pooling слоев следуют полносвязные слои. Fully connected слой принимает все нейроны из предыдущего слоя (будь то fully connected, pooling или convolutional) и соединяет его с каждым нейроном, который он имеет. Fully connected слои не могут быть расположены где угодно, то есть не может быть сверточных слоев после того, как идет fully connected слой



Применение нейронных сетей не ограничивается двумерным случаем. Возможно точно таким же образом построить одно или трехмерные сверточные нейронные сети; convolutional фильтры будут просто иметь размеры определенные надлежащим образом, и pooling слои изменят измерение. Например, возможно использовать одномерные сверточные сети на аудио данных.

# ДЕТЕКТИРОВАНИЕ ЧЕЛОВЕЧЕСКОГО ЛИЦА НА ИЗОБРАЖЕНИИ ДЛЯ ВХОДА СВЕРТОЧНОЙ СЕТИ

Для детектирования лица используется метод, называемый метод Виолы-Джонса, также называемый каскадом Хаара.

Метод был разработан и представлен в 2001 г. Полом Виолой и Майклом Джонсом. Он до сих пор является эффективным методом для поиска объектов на изображениях и видеопоследовательностях в режиме реального времени [1, 2]. Следует отметить, что этот детектор обладает крайне низкой вероятностью ложного обнаружения лица. Метод хорошо работает и обнаруживает черты лица даже при наблюдении объекта под небольшим углом, примерно до 30°. При угле наклона больше 30° вероятность обнаружения лица резко падает. Указанная особенность метода не позволяет в стандартной реализации детектировать лицо человека, повернутое под произвольным углом, что в значительной мере затрудняет или делает невозможным использование алгоритма в современных производственных системах с учетом их растущих потребностей.

1.1. Интегральное представление изображений

Для того, чтобы рассчитать яркость прямоугольного участка изображения, используют интегральное представление [3]. Такое представление используется часто и в других методах, например, в вейвлет преобразованиях, Speeded up robust feature (SURF), фильтрах Хаара и многих разработанных алгоритмах.

Интегральное представление позволяет быстро рассчитывать суммарную яркость произвольного прямоугольника на данном изображении, причем время расчета не зависит от площади прямоугольника. Интегральное представление изображения представляет собой матрицу, совпадающую по размерам с исходным изображением. В каждом ее элементе хранится сумма интенсивностей всех пикселей, находящихся левее и выше данного элемента. Элементы матрицы рассчитываются по следующей формуле:

где – значение точки интегрального изображения; – значение интенсивности исходного изображения. На основе применения интегрального представления изображения вычисление признаков одинакового вида, но с разными геометрическими параметрами, происходит за одинаковое время.

Каждый элемент матрицы представляет собой сумму пикселей в прямоугольнике от до , т. е. значение каждого элемента равно сумме значений всех пикселей левее и выше данного пикселя . Расчет матрицы занимает линейное время, пропорциональное числу пикселей в изображении и его можно производить по следующей формуле:

Интегральное представление имеет интересную особенность. По интегральной матрице можно очень быстро вычислить сумму пикселей произвольного прямоугольника.

1.2. Хаар-подобные характеристики

С точки зрения необходимости использования достаточно простых алгоритмов получения признаков, перспективным является использование Хаар-подобных характеристик, представляющих собой результат сравнения яркостей в двух прямоугольных областях изображения.

Предположим, что задано множество объектов A и множество допустимых ответов B. Пусть g:A→B называется решающей функцией. Решающая функция g должна допускать эффективную компьютерную реализацию, по этой причине её также называют алгоритмом. Признак (feature) f объекта a – отображение f:A→Df где Df – множество допустимых значений признака. В частности, любой алгоритм g:A→B также можно рассматривать как признак. Если задан набор признаков f1,…, fn, то вектор x=(f1(a),…,fn(a)) называется признаковым описанием объекта a∈A. Признаковые описания допустимо отождествлять с самими объектами. При этом множество A=Df1×…×Dfn называют признаковым пространством [4].

В стандартном методе Виолы–Джонса используются прямоугольные признаки, рис. 1. Эти признаки называются примитивами Хаара.

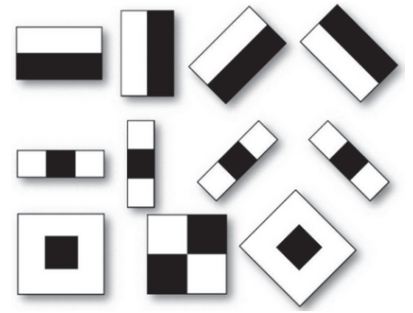


Рис. 1.

Примитивы признаков Хаара

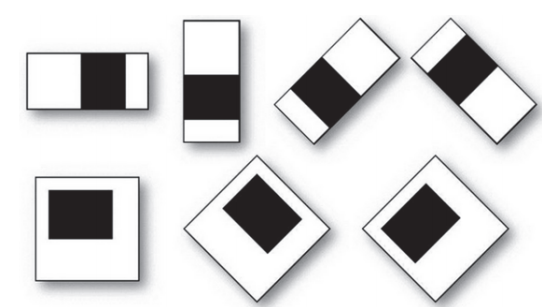


Рис. 2. Дополнительные признаки Хаара

В расширенном методе Виолы–Джонса, представленном в библиотеке OpenCV, используются дополнительные признаки (рис. 2).

Вычисляемым значением такого признака будет:

где – сумма значений яркостей точек, закрываемых светлой частью признака, а – сумма значений яркостей точек, закрываемых темной частью признака. Для их вычисления используется понятие интегрального изображения. Хаар-подобные признаки описывают значение перепада яркости по оси X и Y изображения соответственно.

1.3. Метод построения классификатора на основе алгоритма бустинга

Бустинг – комплекс методов, способствующих повышению точности аналитических моделей. Бустинг (boosting) означает дословно «усиление» «слабых» моделей [5] – это процедура последовательного построения композиции алгоритмов машинного обучения, когда каждый следующий алгоритм стремится компенсировать недостатки композиции всех предыдущих алгоритмов. Идея бустинга была предложена Робертом Шапиро (Schapire) в конце 90 х гг. прошлого века [6], когда надо было найти решение вопроса о том, каким образом имея множество плохих (незначительно отличающихся от случайных) алгоритмов обучения, получить один хороший.

В результате работы алгоритма бустинга на каждой итерации формируется простой классификатор вида:

где – показывает направление знака неравенства; – значение порога; – вычисленное значение признака; z – окно изображения размером 24×24 пикселей. Полученный классификатор имеет минимальную ошибку по отношению к текущим значениям весов, задействованным в процедуре обучения для определения ошибки.

1.4. Метод комбинирования классификаторов в каскадную структуру

Каскадная структура повышает скорость обнаружения, фокусируя свою работу на наиболее информативных областях изображения.

Структура каскадного детектора приведена на рис. 3. Каскад состоит из слоев, которые представляют собой классификаторы, обученные с помощью процедуры бустинга.

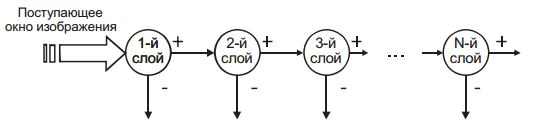


Рис. 3. Структура каскадного детектора

# НАХОЖДЕНИЕ ТОЧЕК ЛИЦА ПО ДЕТЕКТИРОВАННОМУ СЕКТОРУ

# ПРИНЦИП РАБОТЫ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

**Convolution слой**

Предположим, имеется некоторый квадратный слой нейронов. Если использовать фильтр , то выходной сверточный слой будет иметь размер ). Для того чтобы вычислить вход для активационной функции в текущем слое, требуется просуммировать вклады (взвешенных по компонентам фильтров) из клеток предыдущего слоя:

Затем к применятся активационная функция:

Получаем - выход текущего сверточного слоя

**Pooling слой**

Pooling слой сопоставляет региону размера из единственное значение, которое вычисляется по какому-либо принципу. Например max-pooling, берет максимальное значение в этом регионе.

**Fully-connected**

Выход всей сверточной нейронной сети является выходом последнего fully-connected слоя - . Для того чтобы вычислить выходной сигнал от входа применятся некоторая активационная функция ко входу. Исходные данные для первого слоя просто подаются, в то время как входные данные для последующих слоев вычисляются как взвешенные суммы выходных сигналов предыдущего слоя

*Это дает следующий алгоритм прямого распространения для вычисления выхода нейронной сети:*

1. Расчет активации для слоев с известными входами
2. Расчет входов для следующего слоя из предыдущего
3. Повторение шагов 1 и 2, пока не достигнете выходного слоя, то есть значения .

# ПРИНЦИП ОБУЧЕНИЯ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

**Fully-Connected**

Цель обучения – оптимизировать весовые коэффициенты, чтобы минимизировать ошибку. Обучение идет с помощью алгоритма, известного как обратного распространения ошибки. Для того чтобы использовать градиентный спуск (или другой алгоритм) необходимо вычислить производную от ошибки по отношению к каждому весу. Используя цепное правило, получаем, что

Обратим внимание на то, что получаем вклад только от , так как вес не появляется больше нигде. Из уравнения прямого распространения , получаем

Уже известны все значения , так что требуется вычислить частичные производные относительно входного .

Из следует:

это единственное выражение, в котором когда-либо присутствует компонента , следовательно это единственный вклад в правило цепи.

Если (то есть выходной слой), то:

Если же это не выходной слой, то мы просто используем правило цепи еще раз:

*Полный алгоритм обратного распространения:*

1. Вычисление ошибки на выходе слоя
2. Вычисление частной производную от ошибки по отношению к входу нейрона в первом слое, в котором известны ошибки
3. Расчет ошибки на предыдущем слое (ошибки backpropagate):
4. Повторение шагов 2 и 3, Пока дельты известны не на всех слоях, кроме входного.
5. Вычисление градиента ошибки

**Pooling слой**

Получает единственную ошибку, вычисленную от распространения ошибки в обратном направлении от предыдущего слоя. Эта ошибка распространяется на блок, по которому производилась операция pooling.

**Convolutional Layers**

Имеем некоторую функцию ошибки , пришедшую от pooling слоя.

Требуется вычислить . Для этого требуется вычислить градиент для каждого веса путем применения правила цепи.

Требуется суммировать по всем выражениям, в которых есть . Однако известно, что из уравнения прямого распространения.

Для того чтобы вычислить градиент, нужно знать значения . Для этого еще раз используем правило цепи:

Для вычисления весовых коэффициентов для этого сверточного слоя требуется распространять ошибки к предыдущему слою. Еще раз используется правило цепи:

Из уравнения прямого распространения:

Это дает значение ошибки в предыдущем слое. Это похоже на операцию свертки. Есть фильтр , применяемый к слою. Однако, вместо имеем . Приведенное выше выражение имеет смысл только для точек, которые, по крайней мере, отдалены на от верхнего и левого краев. Для того чтобы исправить это, требуется заполнить верхний и левый край нулями. Если сделать это, то это будет операцией свертки , перевернутой по обеим осям.

# МЕТОДЫ ПРИМЕНЯЕМЫЕ ПРИ ОБУЧЕНИИ

**Обучение на GPU**

Важную роль в процессе обучения нейронных сетей занимает время ее обучения, так как современные сети как правило очень сложны и без какой-либо оптимизации они могут обучаться годами. Один из способов ускорить обучение – это распараллеливание процессов.

В сверточных сетях, есть независимые операции(например вычисление по различных фильтрам соответствующие им future map)

В глубоком обучение активно используется обучение на видеокартах, так как они обладают множеством ядер.

Специалисты по обработке и анализу данных как в промышленности, так и в научных кругах используют GPU в сфере машинного обучения, чтобы добиться значительных усовершенствований в широком спектре приложений, включая приложения для классификации изображений, анализа видеоданных, распознавания речи и обработки текстов на естественном языке. Глубокое обучение, то есть использование сложных, многоуровневых нейронных сетей для создания систем, которые могут выявлять признаки из большого объема немаркированных данных, - именно та область, в которой ведутся активные исследования и инвестиционная деятельность.

Хотя машинное обучение существует уже десятки лет, две относительно новые тенденции привели к его широкомасштабному использованию: доступность большого объема данных, а также производительность и эффективность параллельной обработки данных, которая возможна благодаря [вычислениям на GPU](http://www.nvidia.ru/object/gpu-computing-ru.html). GPU используются для обучения этих глубоких нейронных сетей с помощью намного более крупных обучающих последовательностей в более сжатые сроки, с использованием меньшей инфраструктуры ЦОД. GPU также используются, чтобы воспроизводить эти учебные модели машинного обучения для выполнения задач классификации и прогнозирования на облаке. При этом графические процессоры позволяют работать с данными большего объема и с более высокой производительностью, потребляя меньше энергии и на базе меньшей инфраструктуры.

К числу тех, кто впервые применил графические ускорители для решения задач машинного обучения, относятся многие крупные веб-компании и социальные сетевые серверы, наряду с научно-исследовательскими институтами высокого ранга в области обработки и анализа данных и машинного обучения. Благодаря тысячам вычислительных ядер и увеличению производительности приложений в 10-100 раз по сравнению с CPU, GPU стали процессорами, которые выбирают специалисты по обработке данных для работы с данными большого объема.

**Увеличение тестовой выборки**

Зачастую в глубоком обучение наибольшую выгоду приносит простое увеличение тестовой выборки.

**Предотвращение переобучения**

Переобучение, переподгонка (overtraining, overfitting) — нежелательное явление, возникающее при решении задач [обучения по прецедентам](http://www.machinelearning.ru/wiki/index.php?title=%D0%9E%D0%B1%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D0%BE_%D0%BF%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82%D0%B0%D0%BC), когда вероятность ошибки обученного алгоритма на объектах [тестовой выборки](http://www.machinelearning.ru/wiki/index.php?title=%D0%A2%D0%B5%D1%81%D1%82%D0%BE%D0%B2%D0%B0%D1%8F_%D0%B2%D1%8B%D0%B1%D0%BE%D1%80%D0%BA%D0%B0) оказывается существенно выше, чем средняя ошибка на [обучающей выборке](http://www.machinelearning.ru/wiki/index.php?title=%D0%9E%D0%B1%D1%83%D1%87%D0%B0%D1%8E%D1%89%D0%B0%D1%8F_%D0%B2%D1%8B%D0%B1%D0%BE%D1%80%D0%BA%D0%B0). Переобучение возникает при использовании избыточно сложных моделей.

Для того, чтобы избежать чрезмерной подгонки, необходимо использовать дополнительные методы, например:

1. перекрёстная проверка

**Перекрёстная проверка** (англ *Cross-validation*) — метод оценки аналитической модели и её поведения на независимых данных. При оценке модели имеющиеся в наличии данные разбиваются на k частей. Затем на k−1 частях данных производится обучение модели, а оставшаяся часть данных используется для тестирования. Процедура повторяется k раз; в итоге каждая из k частей данных используется для тестирования. В результате получается оценка эффективности выбранной модели с наиболее равномерным использованием имеющихся данных.

1. регуляризация

**Регуляризация** в статистике, машинном обучении, теории обратных задач — метод добавления некоторой дополнительной информации к условию с целью решить некорректно поставленную задачу или предотвратить переобучение. Эта информация часто имеет вид штрафа за сложность модели. Например, это могут быть ограничения гладкости результирующей функции или ограничения по норме векторного пространства.

1. ранняя остановка
2. вербализация нейронных сетей

**Вербализация** — минимизированное описание работы синтезированной и уже обученной нейронной сети в виде нескольких взаимозависимых алгебраических или логических функций.

1. априорная вероятность,
2. байесовское сравнение моделей

которые могут указать, когда дальнейшее обучение больше не ведёт к улучшению оценок параметров. В основе этих методов лежит явное ограничение на сложность моделей, или проверка способности модели к обобщению путём оценки её эффективности на множестве данных, не использовавшихся для обучения и считающихся приближением к реальным данным, к которым модель будет применяться.

**Dropout**

Это техника уменьшения переобучения в нейронных сетях предотвращающая сложную адаптацию к тренировочных данным. Это очень эффективный путь для усреднения результатов нейронной сети. Термин dropout означает исключение некоторого количества случайных нейронов из нейронной сети на текущей итерации.

**Стохастический и пакетный режим обучения**

**Локальные минимумы**

Обратное распространение использует разновидность градиентного спуска, то есть осуществляет спуск вниз по поверхности ошибки, непрерывно подстраивая веса в направлении к минимуму. Поверхность ошибки сложной сети сильно изрезана и состоит из холмов, долин, складок и оврагов в пространстве высокой размерности. Сеть может попасть в локальный минимум (неглубокую долину), когда рядом имеется гораздо более глубокий минимум. В точке локального минимума все направления ведут вверх, и сеть неспособна из него выбраться. Основную трудность при обучении нейронных сетей составляют как раз методы выхода из локальных минимумов: каждый раз выходя из локального минимума снова ищется следующий локальный минимум тем же методом обратного распространения ошибки до тех пор, пока найти из него выход уже не удаётся.

**Перемешивание обучающий примеров**

Так как данные в выборке могут быть изначально в некоторой определенной последовательности(например в выборке данных мужчин и женщин, данные могут быть отсортированы по полу), то получая данные, по порядку их следования, модель лучше обучается по последним данным.

**Нормализация входных данных**

**Паралич сети**

В процессе обучения сети значения весов могут в результате коррекции стать очень большими величинами. Это может привести к тому, что все или большинство нейронов будут функционировать при очень больших значениях OUT, в области, где производная сжимающей функции очень мала. Так как посылаемая обратно в процессе обучения ошибка пропорциональна этой производной, то процесс обучения может практически замереть. В теоретическом отношении эта проблема плохо изучена. Обычно этого избегают уменьшением размера шага η, но это увеличивает время обучения. Различные эвристики использовались для предохранения от паралича или для восстановления после него, но пока что они могут рассматриваться лишь как экспериментальные.

**Локальные минимумы**

**Функция активации**

**Функция потерь**

# РЕЗУЛЬТАТЫ

# ЗАКЛЮЧЕНИЕ

# ЛИТЕРАТУРА

1. Hubel, D. H.; Wiesel, T. N. (1968-03-01)."Receptive fields and functional architecture of monkey striate cortex". *The Journal of Physiology* **195** (1): 215–243.ISSN 0022-3751.PMC 1557912. PMID 4966457.
2. LeCun, Yann; Léon Bottou; Yoshua Bengio; Patrick Haffner (1998)."Gradient-based learning applied to document recognition"
3. <http://www.deeplearningbook.org/>
4. Viola P., Jones M.J. Rapid object detection using a boosted cascade of simple features // IEEE Conf. on Computer Vision and Pattern Recognition. – Kauai, Hawaii, USA, 2001. – V. 1. – P. 511–518. 2.
5. Viola P., Jones M.J. Robust realtime face detection // International Journal of Computer Vision. – 2004. – V. 57. – № 2. – P. 137–154.

# ПРИЛОЖЕНИЕ

Текст программы на языке python