

ГОСУДАРСТВЕННЫЙ КОМИТЕТ СВЯЗИ, ИНФОРМАТИЗАЦИИ И
ТЕЛЕКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ
РЕСПУБЛИКИ УЗБЕКИСТАН

ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ

На правах рукописи
УДК 004.421

АБДУСАТТАРОВ УМИДЖОН ИБРОҲИМЖОН ЎҒЛИ

**Модифицированные методы и алгоритмы распознавание образов при
решении проблем «Data mining»**

5A330201 – Компьютерные системы и их программное обеспечение

Диссертация на соискание академической степени магистра

Научный руководитель:
д.т.н., академик
Камилов М.М.

Ташкент – 2014 г.

**ГОСУДАРСТВЕННЫЙ КОМИТЕТ СВЯЗИ, ИНФОРМАТИЗАЦИИ И
ТЕЛЕКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ
РЕСПУБЛИКИ УЗБЕКИСТАН
ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Факультет: <u>Программный инжиниринг</u>	Студент магистратуры: <u>Абдусаттаров У.И.</u>
Кафедра: <u>Программное обеспечение информационных технологий</u>	Научный руководитель: <u>д.т.н., акад. Камилов М.М.</u>
Учебный год: <u>2012-2014</u>	Специальность: <u>5А330201- "Компьютерные системы и их программное обеспечение"</u>

АННОТАЦИЯ К МАГИСТЕРСКОЙ ДИССЕРТАЦИИ

Актуальность темы. Способность восприятия внешнего мира в форме образов и текста позволяет с определенной достоверностью узнавать бесконечное число объектов на основании ознакомления с конечным их числом, а объективный характер основного свойства образов и текста позволяет моделировать процесс их распознавания.

Цель и задачи исследования. Реализации распознавания текста и лиц по выбранному методу, с анализом возможных улучшений на той или иной стадии процесса. Для достижения поставленной этой цели нам предстояло решить следующие задачи:

- изучить алгоритмы распознавание текста и собрать необходимый теоретический минимум из существующей литературы;
- изучить алгоритм Виолы-Джонса и собрать необходимый теоретический минимум из существующей литературы;
- выбрать среду разработки и реализовать предложенные методы;
- проанализировать результаты проделанной работы;
- сделать обоснованные варианты и предложения об улучшении метода или работы программы.

Объект и предмет исследования. Объектом исследования является распознавание текста с изображения и лиц с видеопотока, предметом исследования назовем сам метод реализации, предложенный Полом Виолой и Майклом Джонсом.

Методы и средства исследования. Для решения поставленных задач используются положения интеллектуального анализа данных, принципы и методы распознавания образов.

Использованные методы. Для достижения поставленной цели использовались методы Пола Виола и Майкла Джонса, метод метрики, алгоритмы распознавания текстов и лиц.

Краткое содержание использованной литературы. Распознаванием текстов и лиц достаточно широко представлено в работах Шоломов Д.Л., Арлазаров В. В., Постников В. В., Шоломов Д.Л., Питерсон У., Уэлдон Э. и многих других ученых. Показано, что одним из эффективных методов разработки прикладного программного обеспечения для распознавания текстов и лиц являются методы Виола-Джонса и метрики.

Научная и практическая значимость. Разработанное автором прикладное программное обеспечение для распознавания текстов с изображений и лиц с видеопотока, благодаря обобщенности, можно использовать не только в качестве основы для дальнейших реализаций подобных ПО, но и в биллинговых системах.

Научная новизна полученная в результате работы заключается в следующем:

- проведен анализ существующих методов по распознаванию текстов и лиц;
 - разработан новый алгоритм на основе метода Виола-Джонса;
- разработан программный продукт в среде (C#) для распознавания текстов с изображений и лиц с видеопотоков.

Состав диссертационной работы. Магистерская диссертация состоит из введения, трех глав, заключения, списка литературы и из трех приложений.

Научный руководитель: _____ Д.т.н., академик Камилов М.М.
Студент магистратуры: _____ Абдусаттаров У.И.

**STATE COMMITTEE FOR COMMUNICATIONS, INFORMATIZATION AND
TELECOMMUNICATION TECHNOLOGIES
OF THE REPUBLIC OF UZBEKISTAN**

TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES

Faculty: <u>Software Engineering</u>	Degree student: <u>Abdusattarov U.I.</u>
Department: <u>The software of information technologies</u>	Scientific supervisor: <u>Ph.D. Kamilov M.M.</u>
	Specialty: <u>5A330201 "Computer systems and their software"</u>

ABSTRACT OF MASTER'S DISSERTATION

Subject topicality. Ability to perceive the outside world in the form of images and text allows a certain authenticity to recognize an infinite number of objects on the basis of acquaintance with a finite number of them, and the objective nature of the main properties of the images and the text allows you to simulate the process of their recognition.

The purpose and objectives of the study. Implementation of OCR, and persons on the selected method, the analysis of possible improvements on the stages of the process. To achieve this goal we had to solve the following tasks:

- OCR algorithms explore and collect the required theoretical minimum of the existing literature;
- explore Viola-Jones algorithm and collect the necessary theoretical minimum of the existing literature;
- choose the programming environment and implement the proposed methods;
- analyze the results of the work done;
- make informed choices and suggestions for improving the method or program.

Object and subject of study. The object of this study is to recognize text from the image and people with video, called himself the subject of research implementation method proposed by Paul Viola and Michael Jones.

Methods and tools for research. To solve the problems using data mining provisions, principles and methods of pattern recognition.

The methods used. To achieve this goal the methods used by Paul Viola and Michael Jones, the metric method, OCR algorithms and individuals.

Summary of literature. OCR, and persons is sufficiently represented in the works Sholomov DL Arlazarov VV, VV Postnikov, Sholomov DL, W. Peterson, E. Weldon and many other scientists. It is shown that one of the effective methods for developing application software for text recognition and those are the methods Viola-Jones and metrics.

Scientific and practical significance. Author developed application software for text recognition with images and those with video, thanks generalization can be used not only as a basis for the further implementation of such software, but also in the billing system.

Scientific novelty of the resulting work is as follows:

- An analysis of existing methods for recognizing texts and persons;
- Developed a new algorithm based on the method of Viola-Jones;
- Developed software environment (C #) for text recognition from images and those with video streams.

The structure of the thesis. The thesis consists of introduction, three chapters, conclusion, references and three software application sections.

Supervisor: _____ Acad., Ph.D. Kamilov M.M.
Graduate student: _____ Abdusattarov U.I.

СОДЕРЖАНИЕ

Введение.....	11
Глава I. Анализ методов и моделей интеллектуального анализа	11
Анализ задач и моделей интеллектуального анализа данных.....	
Анализ методов и алгоритмов интеллектуального анализа данных	4
Анализ существующих методов и алгоритмов распознавания	23
Анализ существующих методов и алгоритмов распознавания лица	25
Выводы по главе I.....	38
Глава II. Модификация методов и алгоритмов распознавания текста и лица	39
Модификация алгоритма распознавания текста с изображения..	39
Модификация алгоритма распознавания лица человека с видеопотока.....	43
Корректировка результатов бесконтактного сканирования и фотографий документов.....	
Выводы по главе II.....	
Глава III. Разработка прикладного программного обеспечения расознавания.....	
Роль и значения программного обеспечения для решения задачи расознавания текстов и лиц	
Назначения, состав и режим работы прикладного программного обеспечения «DPro».....	
Описания модуля «Расознавание текстов».....	

Описания модуля «Распознавания паспорта».....	
Описания модуля «Распознавания лиц».....	
Выводы по главе III.....	
Заключение.....	
Перечень использованной литературы.....	
Приложение.....	

Введение

Актуальность темы. Способность восприятия внешнего мира в форме образов и текста позволяет с определенной достоверностью узнавать бесконечное число объектов на основании ознакомления с конечным их числом, а объективный характер основного свойства образов и текста позволяет моделировать процесс их распознавания.

За последние годы, когда, стремясь к повышению эффективности и прибыльности бизнеса, при создании БД все стали пользоваться средствами обработки цифровой информации, появился и побочный продукт этой активности - горы собранных данных: И вот все больше распространяется идея о том, что эти горы полны золота.

В прошлом процесс добычи золота в горной промышленности состоял из выбора участка земли и дальнейшего ее просеивания большое количество раз. Иногда искатель находил несколько ценных самородков или мог натолкнуться на золотоносную жилу, но в большинстве случаев он вообще ничего не находил и шел дальше к другому многообещающему месту или же вовсе бросал добывать золото, считая это занятие напрасной тратой времени.

Сегодня появились новые научные методы и специализированные инструменты, сделавшие горную промышленность намного более точной и производительной. Data mining для данных развилась почти таким же способом. Старые методы, применявшиеся математиками и статистиками, отнимали много времени, чтобы в результате получить конструктивную и полезную информацию.

В настоящее время на рынке представлено множество инструментов, включающих различные методы, которые делают Data Mining прибыльным делом, все более доступным для большинства компаний.

Термин Data Mining часто переводится как добыча данных, извлечение информации, раскопка данных, интеллектуальный анализ

данных, средства поиска закономерностей, извлечение знаний, анализ шаблонов, "извлечение зерен знаний из гор данных", раскопка знаний в базах данных, информационная проходка данных, "промывание" данных. Понятие "обнаружение знаний в базах данных" (Knowledge Discovery in D

a Понятие Data Mining, появившееся в 1978 году, приобрело высокую популярность в современной трактовке примерно с первой половины 1990-х годов. До этого времени обработка и анализ данных осуществлялся в рамках прикладной статистики, при этом в основном решались задачи обработки небольших баз данных [4].

s О популярности Data Mining говорит и тот факт, что результат поиска ~~КФД) по "Data Mining" в поисковом Data Mining Google~~ (на январь 2014 года) - более 140 миллионов страниц.

Несмотря на то, что в настоящее время большинство документов составляется на компьютерах, задача создания полностью электронного документооборота ещё далека до полной реализации. Как правило, существующие системы охватывают деятельность отдельных организаций, а обмен данными между организациями осуществляется с помощью традиционных бумажных документов.

Задача перевода информации с бумажных на электронные носители является актуальной не только в рамках потребностей, возникающих в системах документооборота, но и в повседневной работе с документами. Современные информационные технологии позволяют нам существенно упростить доступ к информационным ресурсам, накопленным человечеством, при условии, что они будут переведены в электронный вид.

Как отметил Президент И.А.Каримов «Высокими темпами растут высокотехнологичные и рыночные виды услуг, такие как услуги связи и информатизации – на 24,5 процента, по компьютерному программированию – на 18 процентов»[3].

В Республике Узбекистан также ведутся работы в данном направлении: были приняты Закон Республики Узбекистан: в 2004 г. «Об электронном документообороте» [2], в 2003 г «Об информатизации» [1].

Сегодня под определением «распознавание образов» понимается большое количество разнообразных задач, имеющих общий подход, но не имеющих единого универсального решения. Эти задачи встречаются, как и в повседневной жизни, так и в научно-технических вопросах. Создание дифференциального диагноза в медицинской практике, анализ траекторий элементарных частиц в ядерной физике, исследование регулярности в социально-экономической сфере, контроль качества в промышленности, распознавание речи, интеллектуальный видеоконтроль – это лишь малая часть примеров практического использования распознавания.

В данной диссертационной работе рассмотрены распознавания текстов из изображений с использованием метода метрик и распознавания графических образов, а точнее лиц с видеопотока, с использованием метода Виолы-Джонса.

Способность человека при помощи зрения обнаруживать лица и эмоции настолько привычна и проста для нас, что не дает нам повода задуматься, с каким потоком информации справляется наш мозг ежедневно. В то же время, создание подобной компьютерной системы является сложной и недостаточно изученной задачей. Метод Виолы-Джонса является одним из лучших по показателям эффективность/скорость работы и обладает низкой вероятностью ложного обнаружения. Алгоритм распознает объекты при повороте до 30 градусов, но при большем угле качество распознавания падает. Важным также является степень обработки и качество начального изображения, что в первую очередь повлияет на скорость обработки данных.

Поэтому, шагая в ногу со временем и отвечая современным криптографическим требованиям актуально исследовать новые

криптографические системы, применяемые в нашей стране, которые на сегодняшний день малоизучены.

Цель и задачи исследования. Реализации распознавания текста и лиц по выбранному методу, с анализом возможных улучшений на той или иной стадии процесса. Для достижения поставленной этой цели нам предстояло решить следующие задачи:

- изучить алгоритмы распознавание текста и собрать необходимый теоретический минимум из существующей литературы;
- изучить алгоритм Виолы-Джонса и собрать необходимый теоретический минимум из существующей литературы;
- выбрать среду разработки и реализовать предложенные методы;
- проанализировать результаты проделанной работы;
- сделать обоснованные варианты и предложения об улучшении метода или работы программы.

Объект и предмет исследования. Объектом исследования является распознавание текста с изображения и лиц с видеопотока, предметом исследования назовем сам метод реализации, предложенный Полом Виолой и Майклом Джонсом.

Методы и средства исследования. Для решения поставленных задач используются положения интеллектуального анализа данных, принципы и методы распознавания образов.

Использованные методы. Для достижения поставленной цели использовались методы Пола Виола и Майкла Джонса, метод метрики, алгоритмы распознавания текстов и лиц.

Краткое содержание использованной литературы. Распознаванием текстов и лиц достаточно широко представлено в работах Чубукова И. А., Кевин Мерфи, Шоломов Д.Л., Арлазаров В. В., Постников В. В., Шоломов Д.Л., Питерсон У., Уэлдон Э. и многих других ученых. Показано, что одним из эффективных методов разработки прикладного

программного обеспечения для распознавания текстов и лиц являются методы Виола-Джонса и метрики.

Научная и практическая ценность. Разработанное автором прикладное программное обеспечение для распознавания текстов с изображения и лиц с видеопотока, благодаря обобщенности, можно использовать не только в качестве основы для дальнейших реализаций подобных программных обеспечений, но и в биллинговых системах или системах электронного документооборота.

Научная новизна полученная в результате работы заключается в следующем:

проведен анализ существующих методов по распознаванию текстов и лиц;

- разработан новый алгоритм на основе метода Виола-Джонса;

- разработан программный продукт в среде Visual Studio 2013 (C#) для распознавания текстов и паспортов с изображений и лиц с видеопотоков.

Структура и объем работы. Магистерская диссертация состоит из введения, трех глав, заключения, списка литературы, приложения, 2 таблиц и из 44 изображений.

Глава I. Анализ методов и моделей интеллектуального анализа данных

1. Анализ задач и моделей интеллектуального анализа данных

Mining – это сочетание широкого математического инструментария (от классического статистического анализа до новых кибернетических методов) и последних достижений в сфере информационных технологий. В технологии Data Mining гармонично объединились строго формализованные методы и методы неформального анализа, т.е. количественный и качественный анализ данных.

Mining (добыча данных, интеллектуальный анализ данных, глубинный анализ данных) — собирательное название, используемое для обозначения совокупности методов обнаружения в данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности. Термин введен Григорием Пятецким-Шапиро в 1989 году.

По своей сути, интеллектуальный анализ данных – это преобразование больших объемов «сырых» данных в имеющие определенный смысл и практически полезные схемы, структуры и правила. Анализ данных может быть разделен на два вида – прямой (прогнозирование) и косвенный (классификация и кластеризация).

Задача прямого анализа – прогноз конкретных показателей, например, прогноз продажной стоимости дома на базе информации о ценах на дома в данном районе.

Задача косвенного анализа – создание групп данных или поиск определенных структур или схем в существующем наборе данных, например, определение демографической группы типа «Soccer Mom»

(мамы юных футболистов). Каждая перепись населения США подразумевает интеллектуальный анализ данных, так как правительство стремится получить данные о каждом жителе и преобразовать их в информацию, пригодную для дальнейшего практического использования.

Основная цель интеллектуального анализа данных состоит в том, чтобы создать модель, позволяющую эффективно интерпретировать и использовать те данные, которыми вы располагаете сейчас, и те данные, которые вы получите в будущем. Поскольку анализ данных включает в себя целое множество методов, то основной этап создания модели данных – это выбор метода анализа, используемого в этой модели. Для правильного выбора метода требуется практический опыт и некоторое руководство. Далее модель нужно доработать, чтобы сделать ее более эффективной. Познакомившись с материалами этой серии статей, вы сможете определить, какой метод наиболее подходит для вашего набора данных, и какие модификации потребуются для повышения точности модели. Вы сможете самостоятельно создать достаточно эффективную модель для интеллектуального анализа ваших данных.

Задачи, решаемые методами Data Mining, принято разделять на описательные (англ. *descriptive*) и предсказательные (англ. *predictive*).

В описательных задачах самое главное — это дать наглядное описание имеющихся скрытых закономерностей, в то время как в предсказательных задачах на первом плане стоит вопрос о предсказании для тех случаев, для которых данных ещё нет.

К описательным задачам относятся:

- поиск ассоциативных правил или паттернов (образцов);
- группировка объектов, кластерный анализ;
- построение регрессионной модели.

К предсказательным задачам относятся:

- классификация объектов (для заранее заданных классов);

- регрессионный анализ, анализ временных рядов.

Первоначально задача ставится следующим образом:

- имеется достаточно крупная база данных;
- предполагается, что в базе данных находятся некие «скрытые знания».

Необходимо разработать методы обнаружения знаний, скрытых в больших объёмах исходных «сырых» данных. В текущих условиях глобальной конкуренции именно найденные закономерности (знания) могут быть источником дополнительного конкурентного преимущества.

Что означает «скрытые знания»? Это должны быть обязательно знания:

- ранее не известные — то есть такие знания, которые должны быть новыми (а не подтверждающими какие-то ранее полученные сведения);
- нетривиальные — то есть такие, которые нельзя просто так увидеть (при непосредственном визуальном анализе данных или при вычислении простых статистических характеристик);
- практически полезные — то есть такие знания, которые представляют ценность для исследователя или потребителя;
- доступные для интерпретации — то есть такие знания, которые легко представить в наглядной для пользователя форме и легко объяснить в терминах предметной области.

Эти требования во многом определяют суть методов Data mining и то, в каком виде и в каком соотношении в технологии Data mining используются системы управления базами данных, статистические методы анализа и методы искусственного интеллекта.

Интеллектуальный анализ извлекает шаблоны и тренды, существующие в данных. Такие шаблоны и тренды могут быть собраны воедино и определены как модель интеллектуального анализа данных.

Модели интеллектуального анализа данных могут применяться к конкретным бизнес-сценариям, например:

- прогнозирование продаж;
- почтовая рассылка определенным клиентам;
- определение продуктов, которые с высокой долей вероятности могут быть проданы вместе;
- выявление последовательностей в том порядке, в котором клиенты добавляют продукты в корзину для покупок.

Нужно понимать, что построение модели интеллектуального анализа данных является составной частью более масштабного процесса, начинающегося с определения базовой проблемы, которую модель будет решать, и заканчивающегося развертыванием этой модели в рабочей среде.

2. Анализ методов и алгоритмов интеллектуального анализа данных

Основу методов Data Mining составляют всевозможные методы классификации, моделирования и прогнозирования. К методам Data Mining нередко относят статистические методы (дескриптивный анализ, корреляционный и регрессионный анализ, факторный анализ, дисперсионный анализ, компонентный анализ, дискриминантный анализ, анализ временных рядов). Такие методы, однако, предполагают некоторые априорные представления об анализируемых данных, что несколько расходится с целями Data Mining (обнаружение ранее неизвестных нетривиальных и практически полезных знаний).

Одно из важнейших назначений методов Data Mining состоит в наглядном представлении результатов вычислений, что позволяет использовать инструментарий Data Mining людьми, не имеющими специальной математической подготовки. В то же время, применение

статистических методов анализа данных требует хорошего владения теорией вероятностей и математической статистикой.

Знания, добываемые методами Data mining, принято представлять в виде моделей.

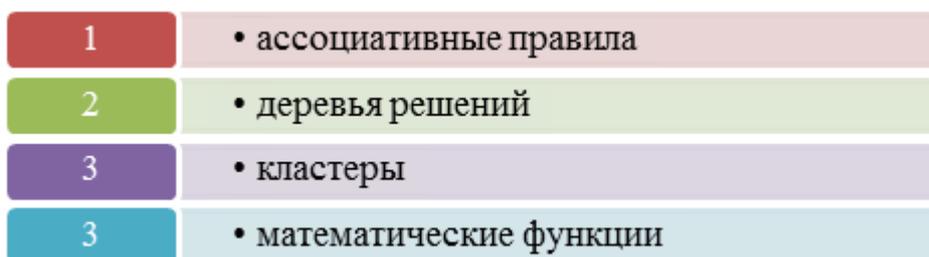


Рис. 1.1 Модели представления знаний Data Mining.

Методы построения таких моделей принято относить к области искусственного интеллекта.

Анализ подходов и методов решения задачи.

Обзор существующих методов

К методам и алгоритмам Data Mining относятся:

1. искусственные нейронные сети;
2. деревья решений, символьные правила;
3. методы ближайшего соседа и k-ближайшего соседа;
4. метод опорных векторов;
- байесовские сети;
6. линейная регрессия;
7. корреляционно-регрессионный анализ;
8. иерархические методы кластерного анализа;
9. неиерархические методы кластерного анализа, в том числе алгоритмы k-средних и k-медианы;
10. методы поиска ассоциативных правил, в том числе алгоритм Apriori;
11. метод ограниченного перебора;
12. эволюционное программирование и генетические алгоритмы;

13. разнообразные методы визуализации данных и множество других методов.

Большинство аналитических методов, используемые в технологии Mining – это известные математические алгоритмы и методы. Новым в их применении является возможность их использования при решении тех или иных конкретных проблем, обусловленная появившимися возможностями технических и программных средств. Следует отметить, что большинство методов Data Mining были разработаны в рамках теории искусственного интеллекта.

Метод представляет собой норму или правило, определенный путь, способ, прием решений задачи теоретического, практического, познавательного, управленческого характера.

Свойства методов Data Mining

Различные методы Data Mining характеризуются определенными свойствами, которые могут быть определяющими при выборе метода анализа данных. Методы можно сравнивать между собой, оценивая характеристики их свойств.

Основные свойства и характеристики методов Data Mining: точность, масштабируемость, интерпретируемость, проверяемость, трудоемкость, гибкость, быстрота и популярность.

Масштабируемость – свойство вычислительной системы, которое обеспечивает предсказуемый рост системных характеристик, например, быстроты реакции, общей производительности и пр., при добавлении к ней вычислительных ресурсов.

В таблице 1 приведена сравнительная характеристика некоторых распространенных методов. Оценка каждой из характеристик проведена следующими категориями, в порядке возрастания: чрезвычайно низкая, очень низкая, низкая/нейтральная, нейтральная/низкая, нейтральная, нейтральная/высокая, высокая, очень высокая.

Таблица 1. Сравнительная характеристика методов Data mining.

АЛГОРИТМ	ТОЧНОСТЬ	МАСШТАБИРУЕМОСТЬ	ИНТЕРПРЕТИРУЕМОСТЬ	ПРИГОДНОСТЬ К ИСП.
линейная регрессия	нейтральная	высокая	высокая / нейтральная	высокая
нейронные сети	высокая	низкая	низкая	низкая
методы визуализации	высокая	очень низкая	высокая	высокая
деревья решений	низкая	высокая	высокая	высокая / нейтральная
нейронные сети	высокая	нейтральная	низкая	высокая / нейтральная
k-ближайшего соседа	низкая	очень низкая	высокая / нейтральная	нейтральная

АЛГОРИТМ	ТРУДОЕМКОСТЬ	РАЗНОСТОРОННОСТЬ	БЫСТРОТА	ПОПУЛЯРНОСТЬ
линейная регрессия	нейтральная	нейтральная	высокая	низкая
нейронные сети	нейтральная	низкая	очень низкая	низкая
методы визуализации	очень высокая	низкая	чрезвычайно низкая	высокая / нейтральная
деревья решений	высокая	высокая	высокая / нейтральная	высокая / нейтральная
нейронные сети	низкая / нейтральная	нейтральная	низкая / нейтральная	нейтральная
k-ближайшего соседа	нейтральная низкая	низкая	высокая	низкая

Как видно из рассмотренной таблицы, каждый из методов имеет свои сильные и слабые стороны. Но ни один метод, какой бы не была его оценка с точки зрения присущих ему характеристик, не может обеспечить решение всего спектра задач Data Mining.

Классификация методов

Работа с данными

Все методы Data Mining можно разделить на две большие группы по принципу работы с исходными обучающими данными. В этой классификации верхний уровень определяется на основании того, сохраняются ли данные после Data Mining либо они дистиллируются для последующего использования.

Непосредственное использование данных, или сохранение данных.

В этом случае исходные данные хранятся в явном детализированном виде и непосредственно используются на стадиях прогностического моделирования и/или анализа исключений. Проблема этой группы методов

– при их использовании могут возникнуть сложности анализа сверхбольших баз данных (Рис. 1.2).

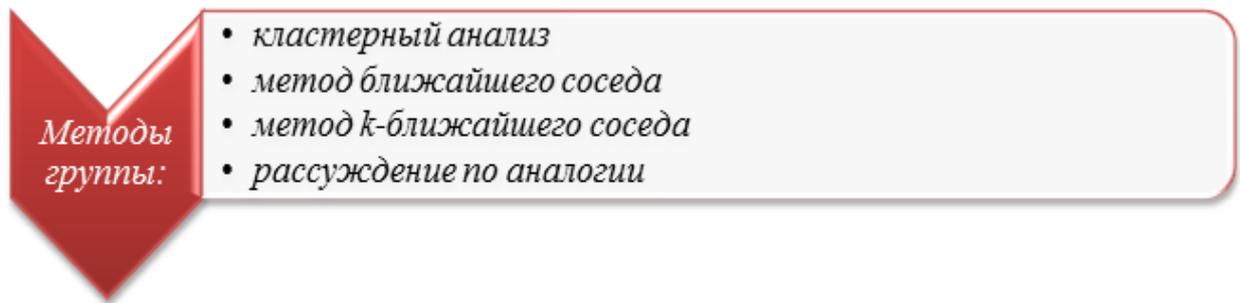


Рис. 1.2 Методы групп.

Выявление и использование формализованных закономерностей, или дистилляция шаблонов.

При технологии дистилляции шаблонов один образец (шаблон) информации извлекается из исходных данных и преобразуется в некие формальные конструкции, вид которых зависит от используемого метода Mining. Этот процесс выполняется на стадии свободного поиска, у первой же группы методов данная стадия в принципе отсутствует. На стадиях прогностического моделирования и анализа исключений используются результаты стадии свободного поиска, они значительно компактнее самих баз данных. Конструкции этих моделей могут быть трактуемыми аналитиком либо не трактуемыми (“черными ящиками”) (Рис. 1.3).

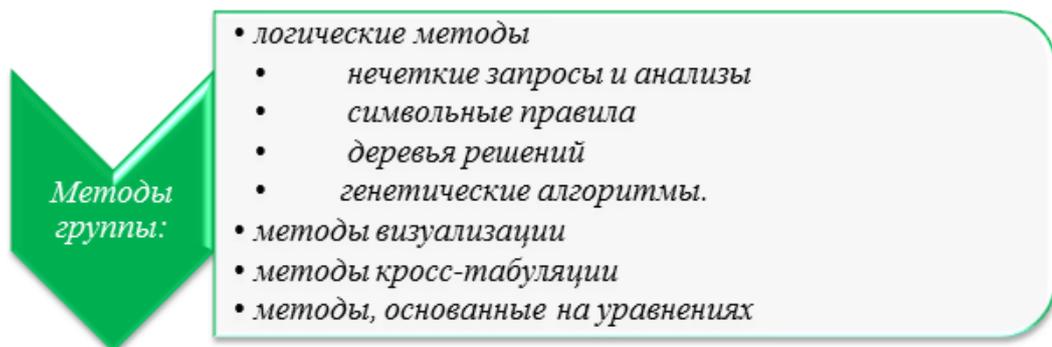


Рис. 1.3 Методы групп.

Логические методы, или методы логической индукции, включают: нечеткие запросы и анализы; символьные правила; деревья решений; генетические алгоритмы.

Методы этой группы являются, пожалуй, наиболее интерпретируемыми – они оформляют найденные закономерности, в большинстве случаев, в достаточно прозрачном виде с точки зрения пользователя. Полученные правила могут включать непрерывные и дискретные переменные. Следует заметить, что деревья решений могут быть легко преобразованы в наборы символьных правил путем генерации одного правила по пути от корня дерева до его терминальной вершины. Деревья решений и правила фактически являются разными способами решения одной задачи и отличаются лишь по своим возможностям. Кроме того, реализация правил осуществляется более медленными алгоритмами, чем индукция деревьев решений.

Методы кросс-табуляции: агенты, байесовские (доверительные) сети, кросс-табличная визуализация.

Последний метод не совсем отвечает одному из свойств Data Mining – самостоятельному поиску закономерностей аналитической системой. Однако предоставление информации в виде кросс-таблиц обеспечивает реализацию основной задачи Data Mining – поиск шаблонов, поэтому этот метод можно также считать одним из методов Data Mining.

Методы на основе уравнений.

Методы этой группы выражают выявленные закономерности в виде математических выражений – уравнений. Следовательно, они могут работать лишь с численными переменными, и переменные других типов должны быть закодированы соответствующим образом. Это несколько ограничивает применение методов данной группы, тем не менее, они широко используются при решении различных задач, особенно задач прогнозирования. Данная классификация разделяет все многообразие

методов Data Mining на две группы: статистические и кибернетические методы.

Подход к обучению математических моделей

Следует отметить, что существует два подхода отнесения статистических методов к Data Mining. Первый из них противопоставляет статистические методы и Data Mining, его сторонники считают классические статистические методы отдельным направлением анализа данных. Согласно второму подходу, статистические методы анализа являются частью математического инструментария Data Mining. Большинство авторитетных источников придерживается второго подхода.

В этой классификации различают две группы методов:

- статистические методы, основанные на использовании усредненного накопленного опыта, который отражен в ретроспективных данных;
- кибернетические методы, включающие множество разнородных математических подходов.

Недостаток такой классификации: и статистические, и кибернетические алгоритмы тем или иным образом опираются на сопоставление статистического опыта с результатами мониторинга текущей ситуации. Преимуществом такой классификации является ее удобство для интерпретации – она используется при описании математических средств современного подхода к извлечению знаний из массивов исходных наблюдений (оперативных и ретроспективных), т.е. в задачах Data Mining.

Статистические методы Data Mining

В эти методы представляют собой четыре взаимосвязанных раздела:

- предварительный анализ природы статистических данных (проверка гипотез стационарности, нормальности, независимости, однородности, оценка вида функции распределения, ее параметров и т.п.);

- выявление связей и закономерностей (линейный и нелинейный регрессионный анализ, корреляционный анализ и др.);
- многомерный статистический анализ (линейный и нелинейный дискриминантный анализ, кластерный анализ, компонентный анализ, факторный анализ и др.);
- динамические модели и прогноз на основе временных рядов.

Арсенал статистических методов Data Mining классифицирован на четыре группы методов (Рис. 1.4):

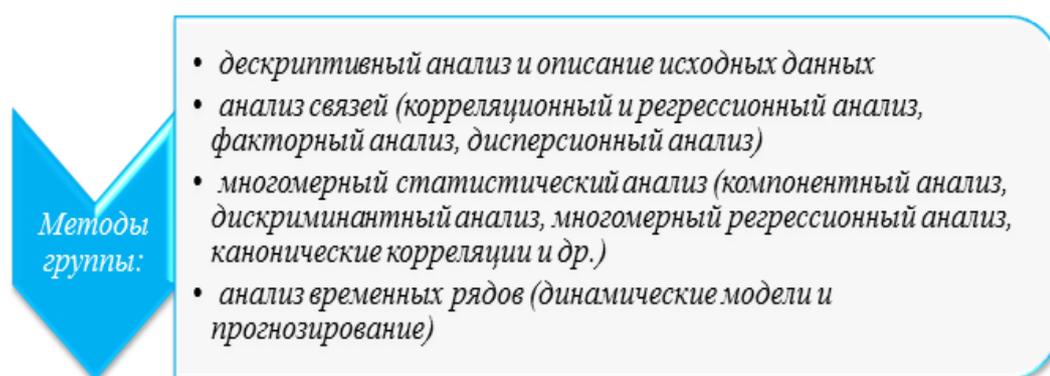


Рис. 1.4 Методы групп.

Кибернетические методы Data Mining

Второе направление Data Mining – это множество подходов, объединенных идеей компьютерной математики и использования теории искусственного интеллекта (Рис. 1.5).

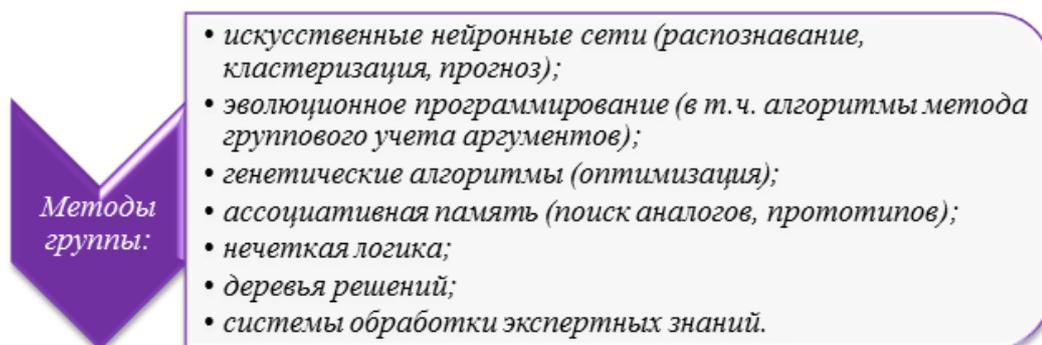


Рис. 1.5 Методы групп.

Классификация по задачам

Методы Data Mining также можно классифицировать по задачам Data Mining. В соответствии с такой классификацией выделяем две группы. Первая из них – это подразделение методов Data Mining на решающие задачи сегментации (т.е. задачи классификации и кластеризации) и задачи прогнозирования. В соответствии со второй классификацией по задачам методы Data Mining могут быть направлены на получение описательных и прогнозирующих результатов.

Описательные методы

Описательные методы служат для нахождения шаблонов или образцов, описывающих данные, которые поддаются интерпретации с точки зрения аналитика (Рис. 1.6).

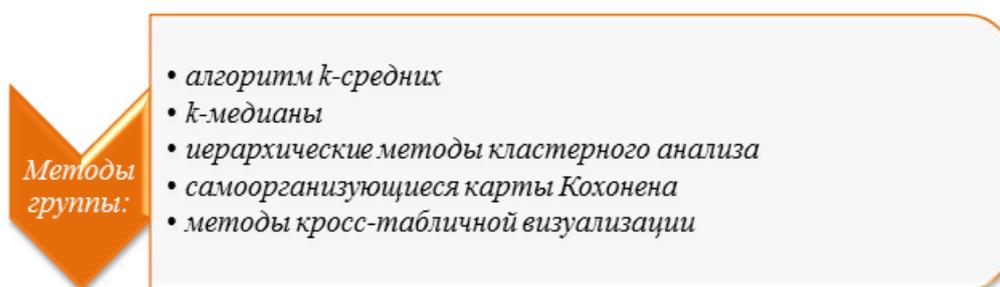


Рис. 1.6 Методы групп.

Прогнозирующие методы

Прогнозирующие методы используют значения одних переменных для предсказания/прогнозирования неизвестных (пропущенных) или будущих значений других (целевых) переменных (Рис. 1.7).

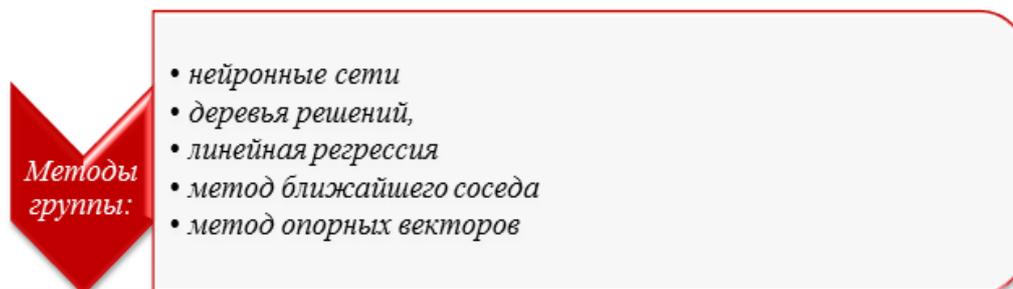


Рис. 1.7 Методы групп.

3. Анализ существующих методов и алгоритмов распознавания текста

В современной постановке задача распознавания стоит, прежде всего, для так называемых бизнес-форм. То есть документов коммерческого свойства, прежде всего финансовых. Примером бизнес-формы может служить товарная накладная, которая является одним из основных видов документов, используемых в торговле.

Для бизнес-форм характерна неравнозначность информации, расположенной в различных полях формы. Прежде всего, важными полями являются поля сумм, номеров счетов и т.п. Примером критически важного поля может служить номер паспорта в форме, где используются паспортные данные.

Для повышения качества распознавания критически важных полей форм применяются различные методы. В частности, используются методы с введением в данные дополнительной избыточной информации. Широко известным примером такого метода из области теории информации являются коды Хемминга [7]. Ряд методов в области распознавания текстов предложен в работе [11].

В этой части диссертационной работы рассмотрен универсальный алгоритм «рулетки» для распознавания полей с проверочной функцией.

Применительно к задаче распознавания существует класс полей, содержащий в самой своей структуре дополнительную информацию, которая может служить для проверки корректности распознавания. А также для исправления ошибок, если ставится такая задача.

Можно разделить использование дополнительной информации условно на два типа – корректирующая и отбраковывающая проверки. Для отбраковывающей проверки характерно использование заранее предопределённых значений на соответствие (например, широко распространённая проверка по словарю). В этом случае при отсутствии

полученного при распознавании значения в словаре, мы выносим решение о некорректности распознавания.

Корректирующая проверка отличается от отбраковывающей тем, что мы можем попытаться восстановить неправильно распознанное значение.

Для каждого символа существуют альтернативы распознавания. Можно проверить значение, заменив один (или несколько) символ на его альтернативу. Такой метод, примененный к значению без контрольных данных значительно менее результативен – так как мы фактически пытаемся угадать, что же было распознано неправильно. В силу того, что вероятность ошибки, прежде всего, зависит от самого символа, то сделать однозначный вывод о том, какой из символов был распознан некорректно исходя из общих соображений нельзя. С другой стороны, имея контрольную информацию, мы можем проверить корректность замены символа на его альтернативу.

В силу того, что алгоритм контрольного значения выбирается с тем расчётом, чтобы близкие значения основных данных соответствовали существенно отличным контрольным данным, и, принимая во внимание малую вероятность ошибки, мы приходим к тому, что может восстановить изначальные данные с большой долей уверенности.

Такая дополнительная информация может быть выражена в любой форме, но наибольшее распространение получили так называемые контрольные суммы.

Математическая постановка задачи

Задача распознавания с коррекцией сводится к перебору элементов

в
е
к
т
о
р
а

$$T\left(\left\{a_i^k\right\}_{i=1}^n\right)=\begin{cases} 0, & \text{если тест не пройден;} \\ 1, & \text{если тест пройден.} \end{cases}$$

а

Общее количество возможных интерпретаций задается формулой

$$\prod_{i=1}^n |\bar{a}_i|$$

Г

д Уже для 2 вариантов для каждого символа слова длиной 15 символов эта формула даёт 32768 вариантов интерпретации, что, при достаточно сложной функции проверки T , может приводить к длительным задержкам при распознавании. Но, как показывает опыт практического применения, большая часть слов распознается при проверке одного варианта для каждого символа, т.е. для слова длиной символов нужно рассмотреть всего лишь порядка 15 вариантов распознавания.

и *Алгоритм корректировки*

ч Алгоритм, предлагаемый для отбраковки и/или восстановления данных с контрольными значениями.

с В силу того, что вероятность ошибки в любом символе одинакова, алгоритмом не делается различие контрольных и ординарных разрядов. Алгоритм последовательно сменяет альтернативы, комбинируя их для всех символов до тех пор, пока комбинация альтернатив не удовлетворит используемой проверке. В силу сложности алгоритма проверки контрольного разряда существует возможность существенно понизить вероятность неверного распознавания.

м

в 4. Анализ существующих методов и алгоритмов распознавания

о

лица

л

Человек даже не замечает, как он просто справляется с задачами обнаружения лиц и эмоций при помощи своего зрения. Когда глаз смотрит на окружающие лица людей, предметы, природу, подсознательно не чувствуется, какой объем работы прodelывает мозг, чтобы обработать весь поток визуальной информации. Человеку не составит труда найти

с

л

о

знакомого человека на фотографии, или отличить ехидную гримасу от улыбки.

Человек пытается воссоздать и построить компьютерную систему обнаружения лиц и эмоций — ему это отчасти удается, но каждый раз приходится сталкиваться с большими проблемами при распознавании. Компьютеры в наше время беспрепятственно могут хранить огромные объемы информации, картинки, видео и аудио файлы. Но отыскать вычислительным системам с такой же легкостью, к примеру, нужную фотографию с определенной эмоцией нужного человека из собственной личной фотогалереи — сложная задача.

Решению такой задачи мешают некоторые факторы:

- Разный размер искомых объектов, а также масштаб изображений;
- Определяемый объект может находиться где угодно на изображении;
- Совершенно другой объект может быть похож на искомый;
- Предмет, который мы воспринимаем как что-то отдельное, на изображении никак не выделен, и находится на фоне других предметов, сливается с ними;
- Старые и необработанные фотографии — на них всегда присутствуют «отвлекающие» систему царапины, помехи, искажения, на сканируемых фото не редко появляются разного рода муары;
- Не стоит забывать, что во многих алгоритмах распознавания (также и в Виола-Джонс) работа идет с 2D-пространстве непосредственно. Поэтому поворот искомого объекта и изменение угла обзора относительно заданных координатных осей проекции влияют на его проекцию в 2D. Один и тот же объект может давать совершенно разную картинку, в зависимости от поворота или расстояния до него. Искомое лицо может быть повернуто в плоскости изображения. Даже относительно небольшое изменение ориентации лица относительно камеры влечет за собой

серьезное изменение изображения лица и о распознании мимики данного лица уже и речи быть не может;

- Качество изображения или кадра: за светом и неправильный баланс белого, цветокоррекция и другие параметры, безусловно, влияют на распознавание объекта;

- Расовая принадлежность людей: цвет кожи, расположение и размеры отдельных распознаваемых признаков;

- Сильное изменение выражения лица. Например, чересчур показное действие может сильно оказать влияние на правильное распознавание определенной эмоции;

- Индивидуальные особенности лица человека, такие как усы, борода, очки, морщины, существенно осложняют автоматическое распознавание;

- Часть лица вообще может быть невидима или обрезана;

- Лица может не быть совсем на фотографии, но машина, как ей кажется, правильно определяет другие объекты за лицо и черты лица и детектирует именно их.

- Список можно продолжать еще долго. Но внимание заострено на самых важных моментах, поэтому перечислять все мешающие параметры нет смысла.

Сравнение качества распознавания разнообразных методов осложнено многими причинами. Одна из них, и самая весомая – это то, что в большинстве случаев опираться можно только на данные испытаний, предоставляемые самими авторами, так как проведение крупномасштабного исследования по реализации большинства известных методов и сравнения их между собой на едином наборе изображений не представляется возможным:

- необходима универсальная коллекция тестовых данных;

- должны присутствовать одинаковые наборы данных;

- необходимы вычислительная мощность — ресурсы уровня одной лаборатории для этого малы;
- высокая трудоемкость исследования данных алгоритмов;
- на основе информации, предоставляемой авторами методов, также сложно провести корректное сравнение, поскольку проверка методов часто производится на разных наборах изображений, с разной формулировкой условий успешного и неуспешного обнаружения. К тому же проверка для многих методов первой категории производилась на значительно меньших наборах изображений.

Алгоритм Виолы—Джонса для распознавания лиц

Алгоритм Виолы—Джонса (англ. Viola–Jones object detection) — алгоритм, позволяющий обнаруживать объекты на изображениях в реальном времени. Предложен в 2001 году Пол Виола (Paul Viola) и Майкл Джонс (Michael Jones). Хотя алгоритм может распознавать различные классы изображений, основной задачей при его создании было обнаружение лиц. Существует множество реализаций, в том числе в составе библиотеки компьютерного зрения OpenCV (функция `cvHaarDetectObjects()`).

Данный алгоритм до сих пор является основополагающим для поиска объектов на изображении в реальном времени. Алгоритм находит лица с высокой точностью и низким количеством ложных срабатываний.

Общая схема метода Виолы-Джонса

Алгоритм распознавания по методу Виолы-Джонса основан на "суммировании" пикселей (с определенными весовыми коэффициентами) под скользящим [по растру] окном. Распознавание в этом методе осуществляется по "прецедентам". С помощью "обучающей выборки" строится набор "сильных классификаторов", каждый из которых для квадратного окна говорит: "предположительно, в окне - лицо", или - "определенно, не лицо". Таким образом, для того, чтобы алгоритм признал картинку в окне за лицо, необходимо, чтобы все "сильные

классификаторы" (stages) ответили: "да, лицо предположительно есть". Если хотя бы один из них отверг окно (сказал, что "лица определенно нет"), то алгоритм сразу же отвергает данное окно, другие "сильные классификаторы" не использует, и переходит к следующему окну.

Описание метода Viola Jones

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 лицо.

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 эффективность распознавания скорость работы. Также этот детектор обладает крайне низкой вероятностью ложного обнаружения лица. Алгоритм даже хорошо работает и распознает черты лица под небольшим углом, примерно до 30 градусов. При угле наклона больше 30 градусов процент обнаружений резко падает. И это не позволяет в стандартной реализации детектировать повернутое лицо человека под произвольным углом, что в значительной мере затрудняет или делает невозможным использование алгоритма в современных производственных системах с учетом их растущих потребностей. Требуется подробный разбор принципов, на которых

основан алгоритм Виолы-Джонса. Данный метод в общем виде ищет лица и черты лица по общему принципу сканирующего окна.

Принцип сканирующего окна

В общем виде, задача обнаружения лица и черт лица человека 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 255^3 , если это цветное изображение (компоненты R, G,

- В результате своей работы, алгоритм должен определить лица и их черты и пометить их – поиск осуществляется в активной области изображения прямоугольными признаками, с помощью которых и описывается найденное лицо и его черты (1.1):

$$rectangle_i = \{x, y, w, h, a\}, \quad (1.1)$$

где x, y – координаты центра i -го прямоугольника, w – ширина, h – высота, a – угол наклона прямоугольника к вертикальной оси изображения.

Иными словами, применительно к рисункам и фотографиям используется подход на основе сканирующего окна (scanning window): сканируется изображение окном поиска (так называемое, окно сканирования), а затем применяется классификатор к каждому положению. Система обучения и выбора наиболее значимых признаков полностью автоматизирована и не требует вмешательства человека, поэтому данный подход работает быстро.

Задача поиска и нахождения лиц на изображении с помощью данного принципа часто бывает очередным шагом на пути к распознаванию

характерных черт, к примеру, верификации человека по распознанному лицу или распознавания мимики лица.

Интегральное представление изображений

Для того, чтобы производить какие-либо действия с данными, используется интегральное представление изображений [3] 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. Элементы матрицы рассчитываются по следующей формуле (1.2):

$$L(x,y) = \sum_{i=0}^x \sum_{j=0}^y I(i,j), \quad (1.2)$$

где $I(i,j)$ — яркость пикселя исходного изображения.

Каждый элемент матрицы $L[x,y]$ представляет собой сумму пикселей в прямоугольнике от $(0,0)$ до (x,y) , т.е. значение каждого пикселя (x,y) равно сумме значений всех пикселей левее и выше данного пикселя (x,y) . Расчет матрицы занимает линейное время, пропорциональное числу пикселей в изображении, поэтому интегральное изображение просчитывается за один проход.

Расчет матрицы возможен по формуле (1.3):

$$L(x,y) = I(x,y) - L(x-1,y-1) + L(x,y-1) + L(x-1,y) \quad (1.3)$$

По такой интегральной матрице можно очень быстро вычислить сумму пикселей произвольного прямоугольника, произвольной площади.

Пусть в прямоугольнике ABCD есть интересующий нас объект D (Рис. 1.8):

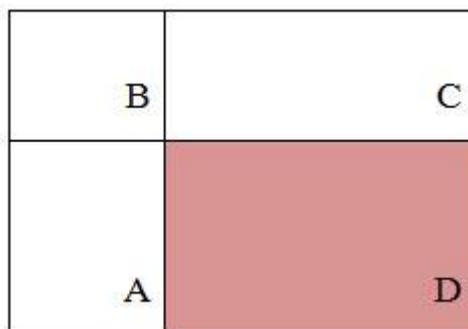


Рис. 1.8 Прямоугольник ABCD.

Из рисунка (Рис. 1.8) понятно, что сумму внутри прямоугольника можно выразить через суммы и разности смежных прямоугольников по следующей формуле(1.4):

$$S(ABCD) = L(A) + L(C) - L(B) - L(D) \quad (1.4)$$

Признаки Хаара

Признак — отображение $f: X \Rightarrow D_f$, где D_f — множество допустимых значений признака. Если заданы признаки f_1, \dots, f_n , то вектор признаков $x = (x_1, \dots, x_n)$ называется признаковым описанием объекта $x \in X$. Признаковые описания допустимо отождествлять с самими объектами. При этом множество $X = D_{f_1} * \dots * D_{f_n}$ называют признаковым пространством [1].

Признаки делятся на следующие типы в зависимости от множества

- бинарный признак, $D_f = \{0,1\}$;
- номинальный признак: D_f — конечное множество;

- порядковый признак: D_f — конечное упорядоченное множество;
- количественный признак: D_f — множество действительных чисел.

Естественно, бывают прикладные задачи с разнотипными признаками, для их решения подходят далеко не все методы.

В стандартном методе Виолы–Джонса используются прямоугольные признаки, изображенные на рисунке ниже, они называются примитивами Хаара (Рис. 1.9):

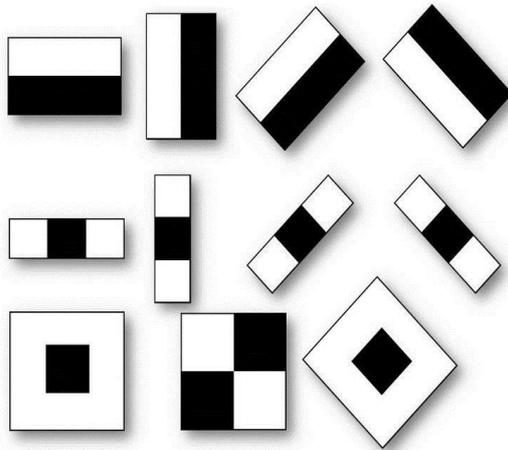


Рис. 1.9 Примитивы Хаара.

В расширенном методе Виолы – Джонса, используемом в библиотеке OpenCV используются дополнительные признаки:

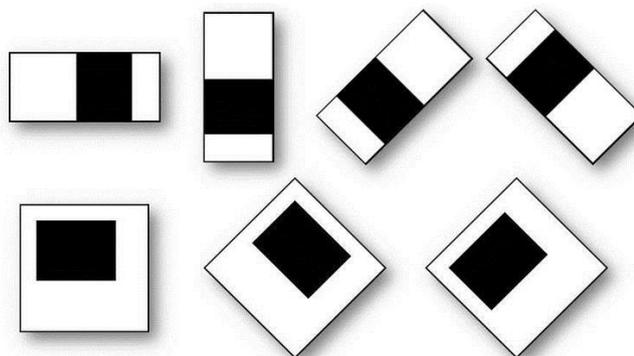


Рис. 1.10 Дополнительные признаки Хаара.

Вычисляемым значением такого признака будет

X

где X – сумма значений яркостей точек закрываемых светлой частью признака, а Y – сумма значений яркостей точек закрываемых темной частью признака. Для их вычисления используется понятие интегрального изображения, рассмотренное выше.

Признаки Хаара дают точечное значение перепада яркости по оси X и Y соответственно.

Сканирование окна

Алгоритм сканирования окна с признаками выглядит так:

- есть исследуемое изображение, выбрано окно сканирования, выбраны используемые признаки;
- далее окно сканирования начинает последовательно двигаться по изображению с шагом в 1 ячейку окна (допустим, размер самого окна есть $24*24$ ячейки);
- при сканировании изображения в каждом окне вычисляется приблизительно 200 000 вариантов расположения признаков, за счет изменения масштаба признаков и их положения в окне сканирования;
- сканирование производится последовательно для различных масштабов;
- масштабируется не само изображение, а сканирующее окно (изменяется размер ячейки);
- все найденные признаки попадают к классификатору, который «выносит вердикт».

В процессе поиска вычислять все признаки на маломощных настольных ПК просто нереально. Следовательно, классификатор должен реагировать только на определенное, нужное подмножество всех признаков. Совершенно логично, что надо обучить классификатор

нахождению лиц по данному определенному подмножеству. Это можно сделать, обучая вычислительную машину автоматически.

Используемая в алгоритме модель машинного обучения

Обучение машины — это процесс получения модулем новых знаний. Есть признанное определение данному процессу:

«Машинное обучение — это наука, изучающая компьютерные алгоритмы, автоматически улучшающиеся во время работы» (Michel, 1996)

Ниже показан процесс обучения машины (Рис 1.11):

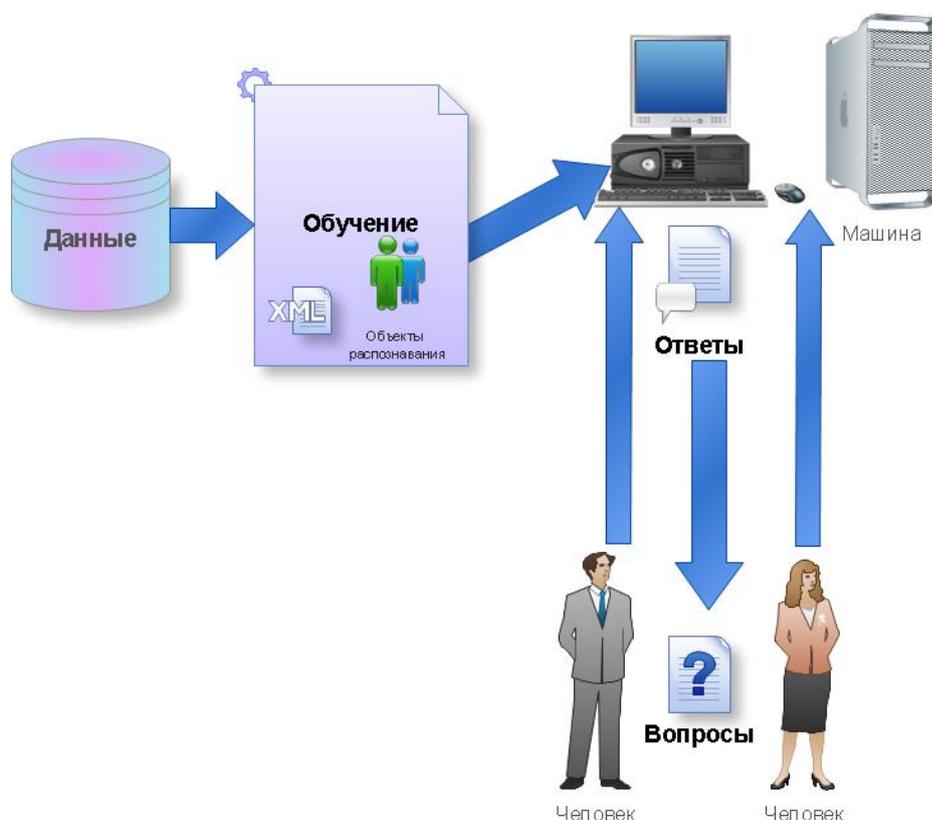


Рис. 1.11 Обучение машины.

Данный процесс входит в концепцию и технологию под названием mining (извлечение информации и интеллектуальный анализ данных), куда входят помимо Машинного обучения такие дисциплины, как Теория баз данных, Искусственный интеллект, Алгоритмизация, Распознавание образов и прочие.

Машинное обучение в методе Виолы-Джонса решает такую задачу как классификация.

Обучение классификатора в методе Виолы-Джонса

В контексте алгоритма, имеется множество объектов (изображений), разделённых некоторым образом на классы. Задано конечное множество изображений, для которых известно, к какому классу они относятся (к примеру, это может быть класс «фронтальное положение носа»). Это множество называется обучающей выборкой. Классовая принадлежность остальных объектов не известна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества.

Классифицировать объект — значит, указать номер (или наименование класса), к которому относится данный объект.

Классификация объекта — номер или наименование класса, выдаваемые алгоритмом классификации в результате его применения к данному конкретному объекту.

Классификатор(classifier) — в задачах классификации это аппроксимирующая функция, выносящая решение, к какому именно классу данный объект принадлежит.

Обучающая выборка – конечное число данных.

В машинном обучении задача классификации относится к разделу обучения с учителем когда классы поделены. Распознавание образов по сути своей и есть классификация изображений и сигналов. В случае алгоритма Виолы-Джонса для идентификации и распознавания лица классификация является двухклассовой.

Постановка классификации выглядит следующим образом:

Есть X – множество, в котором хранится описание объектов, Y – конечное множество номеров, принадлежащих классам. Между ними есть зависимость – отображение $Y^*: X \Rightarrow Y$. Обучающая выборка представлена признаков X , которая выдает ответ для любого возможного наблюдения X и

способна классифицировать объект $x \in X$. Данное простое правило должно хорошо работать и на новых данных.

Принципы решающего дерева в разрабатываемом алгоритме

Дерево принятия решений — это дерево, в листьях которого стоят значения целевой функции, а в остальных узлах — условия перехода (к примеру, на Лице есть Улыбка), определяющие по какому из ребер идти. Если для данного наблюдения условие равно истине то осуществляется переход по левому ребру, если же ложь — по правому [4]. Для примера, дерево представлено на следующем рисунке (Рис 1.12):

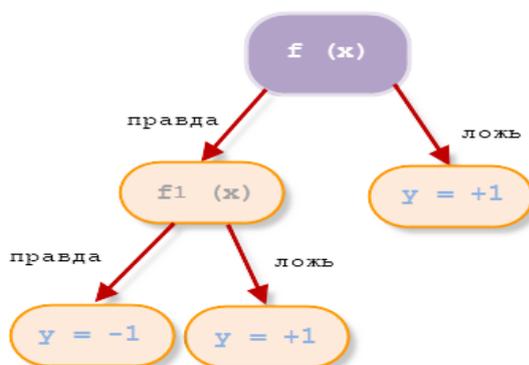


Рис. 1.12 Дерево принятия решений.

Достоинствами таких решающих деревьев являются наглядность, легкость работы с ними, быстроедействие. Также, они легко применяются для задач с множеством классов.

Выводы по главе I

1. Изучены и проанализированы виды и методы групп интеллектуального анализа данных.

Изучены и проанализированы существующие методы распознавания текстов и лиц.

Исследованы и проанализированы использованные методы в существующих алгоритмах.

4. Изучена и приведена общая постановки задачи распознавания текстов и лиц.

Глава II. Модификация методов и алгоритмов распознавания текста и лица

1. Модификация алгоритма распознавания текста с изображения

Распознавание образов или теория распознавания образов, это раздел информатики и смежных дисциплин, развивающий основы и методы классификации и идентификации предметов, явлений, процессов, сигналов, ситуаций и т. п. объектов, которые характеризуются конечным набором некоторых свойств и признаков. Данное определение нам дает

Также она утверждает, что можно выделить два основных направления

- Изучение способностей к распознаванию, которыми обладают живые существа, объяснение и моделирование их;
- Развитие теории и методов построения устройств, предназначенных для решения отдельных задач в прикладных целях.

Моя работа в рамках диссертационной работы проводилась именно в направлении второго пункта.

Итак, первая — это распознавание текста на графических изображениях и сейчас говорить о важности данного подраздела не приходится. Всем давно известно, что существуют миллионы старых книг, которые хранятся в хранилищах, доступ к которым имеет только специализированный персонал. Использование этих книг запрещено по причине их ветшалости и дряхлости, так как возможно, что они могут рассыпаться прямо в руках читателя, но знания, которые они хранят, представляют, несомненно, большой клад для человечества и поэтому оцифровка этих книг столь важна. Именно этим в частности занимаются специалисты в области обработки данных.

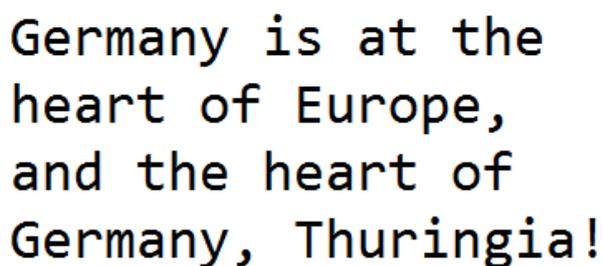
Следует заметить, что под распознаванием текста обычно понимают три главных метода.

- Сравнение с заранее подготовленным шаблоном;
- Распознавание с использованием критериев, распознаваемого объекта;
- Распознавание при помощи самообучающихся алгоритмов, в том числе при помощи нейронных сетей.

Также следует сказать, что распознавание текста почти всегда идет в купе с обнаружением текста на изображении, но так как я не ставил этой цели, этап обнаружения был опущен и заменен на легкую предобработку.

Теперь о самой работе. Было написано приложение, способное распознавать текст при использовании изображений высокого либо среднего качества, со слабым шумом либо без него. Приложение способно распознавать буквы английского и русского алфавита, верхнего и нижнего регистра. Изображение подается для распознавания непосредственно из самого приложения.

Так как этап обнаружения был опущен и вставлен этап предобработки, то изображение в большинстве своем выглядит следующим образом.



Germany is at the
heart of Europe,
and the heart of
Germany, Thuringia!

Рис. 2.1 Выявления распознавания текста.

Данное изображение обрабатывается двумя фильтрами: Медианным и монохромом (Приложение). В приложении прикладного программного

обеспечения использовалась измененная версия медианного фильтра с увеличением значения компоненты красного цвета.

Медианный фильтр применяется для минимизации шума и смазывания острых краев букв (засечек и т.п.). После этого изображение обрабатывается монохромом. То есть происходит четкая бинаризация, при этом границы букв четко фиксируются.

Сегментация

После предобработки в процессе распознавания происходит сегментация изображения. Опять-таки, так как этап обнаружения опущен, то для процесса сегментации принята следующая эвристика. Предполагается, что предложения текста расположены горизонтально и не создают пересечений друг с другом. Тогда задача сегментации не составляет труда.

Задается среднее значение расстояния между двумя буквами в слове. После этого изображение делится на строки путем поиска полных белых полос. Далее эти полосы делятся на слова путем поиска белых полос определенной ширины. После всего этого выделенные слова передаются на заключительный этап, и они делятся на буквы. Таким образом, на выходе модуля сегментации мы имеем весь текст представленный изображениями букв этого текста.

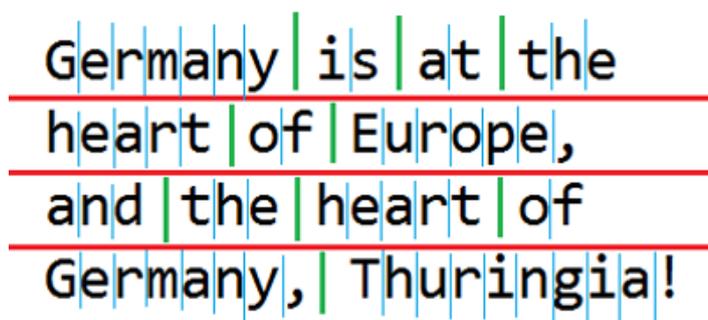


Рис. 2.2 Выявление распознавания текста.

Непосредственно перед распознаванием изображение нормализуется и приводится до размеров шаблонов, подготовленных заранее.

Распознавание

Метрика – некоторое условное значение функции, определяющее положение объекта в пространстве. Таким образом, если два объекта расположены близко друг от друга, то есть похожи (например, две буквы А написанные разным шрифтом), то метрики для таких объектов будут совпадать или быть предельно похожими. Для распознавания в этом режиме была выбрана метрика Хэмминга.

Метрика Хэмминга – метрика, которая показывает, как сильно объекты не похожи между собой.

Данную метрику часто используют при кодировании информации и передаче данных. Например, после сеанса передачи на выходе имеется следующая последовательность бит (1001001), также нам известно, что должна прийти другая последовательность бит (1000101). Мы вычисляем метрику путем сравнения частей последовательности с соответствующими местами из другой последовательности. Таким образом, метрика Хэмминга в нашем случае равна 2. Так как объекты отличаются в двух позициях. 2-это степень непохожести, чем больше, тем хуже в нашем случае.

Следовательно, чтобы определить, какая буква изображена, нужно найти ее метрику со всеми готовыми шаблонами. И тот шаблон, чья метрика окажется наиболее близкой к 0 будет ответом.

Но как показала практика, подсчет одной лишь метрики не дает положительного результата, так многие буквы похожи между собой. Например «j» «i», что приводит к ошибочному распознаванию.

Тогда было принято решение придумать новые метрики, позволяющие разграничить некоторое множество букв в отдельный класс. В частности, были реализованы метрики (Отражения горизонтального и вертикального, преобладания веса горизонтального и вертикального).

Экспериментом было выяснено, что такие буквы как «Н» «I» «i» «O» «l» обладают суперсимметрией (полностью совпадают со своими отражениями и значимые пиксели распределены равномерно по всему изображению), поэтому они были вынесены в отдельный класс, что сокращает перебор всех метрик примерно в 6 раз. Аналогичные действия были проведены в отношении других букв. В среднем уменьшение перебора достигает примерно 3 раза.

Также есть уникальная буква такая как «J», которая находится в своем классе одна, и значит, идентифицируются однозначно. Далее, для каждого класса высчитывается метрика Хэмминга, которая на данном этапе дает лучшие показатели, чем при прямом применении.

При создании шаблонов использовался шрифт «Arial», поэтому, если распознаваемый текст написан этим шрифтом, распознавание имеет точность порядка 99 процентов. При изменении шрифта, точность падает до 70 процентов.

2. Модификация алгоритма распознавания лица человека с видеопотока

Обучение классификаторов идет очень медленно, но результаты поиска лица очень быстры. Виола-Джонс является одним из лучших по соотношению показателей эффективности распознавания/скорость работы. Также этот детектор обладает крайне низкой вероятностью ложного обнаружения лица. Алгоритм даже хорошо работает и распознает черты лица под небольшим углом, примерно до 30 градусов. Однако при угле наклона больше 30 градусов процент обнаружений резко падает.

Предлагаемый ниже метод предполагает возможное избавление от:

- больших ограничений в виде недостаточной освещенности, существующих помех на изображении и неразличимого объекта на фоне с помощью предварительной обработки изображения;

- проблемы угла наклона лица путем тренировки новых каскадов, специально обученных на нахождение наклоненной головы;
- неточностей обнаружения эмоций на лице человека путем перечисленных выше пунктов, путем введения новых примитивов Хаара, расширяющих стандартный набор, реализуемый в алгоритме Виолы-Джонса и путем обучения большого количества каскадов, специально заточенных под то или иное состояние черты лица определенной эмоции.

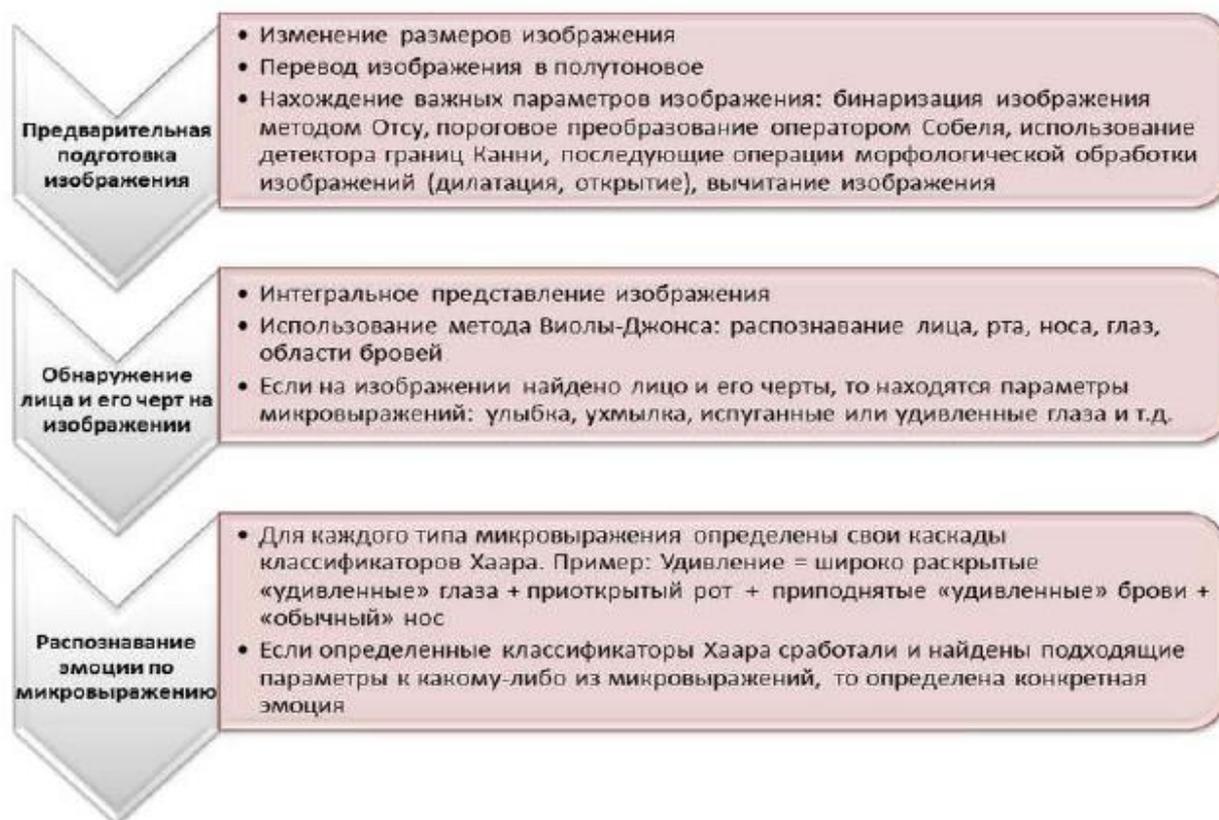


Рис. 2.3 Этапы распознавания лица.

Предварительная обработка изображения.

Для увеличения процента распознавания с помощью алгоритма Виолы - Джонса, с целью увеличения производительности необходимо выполнять предварительную обработку изображения. Для создания быстрого и надежного способа определения вероятных областей лица человека с целью ускорения обработки на дальнейших этапах обнаружения, предлагается алгоритм определения и выделения граничных

контуров. Идея данного подхода заключается в том, что можно подчеркнуть те области, в которых с наибольшей вероятностью можно будет найти лицо и его черты. Тем самым достигается не только ускорение работы алгоритма, но и уменьшается вероятность ложных обнаружений лиц.

Изображение в градациях серого

Для начала нужно перевести изображение в градации серого (Рис. Для этого удобно представить изображение в цветовой модели YUV. Для этого выполняется конверсия:



Рис. 2.4 Конвертация цветного изображения в градациях серого.

$$\begin{aligned} \bullet Y &= 0.299 * R + 0.587 * G + 0.114 * B; \\ \bullet U &= -0.14713 * R - 0.28886 * G + 0.436 * B; \\ \bullet V &= 0.615 * R - 0.51499 * G - 0.10001 * B; \end{aligned} \quad (2.1)$$

где R, G и B – это интенсивности заданных цветов, если совсем точно, то это матрицы, описывающие компоненты модели (R,G,B). Y в формуле – это яркостная составляющая, а U и V — цветоразностные составляющие, так называемые сигналы цветности (эти три параметра тоже представлены матрицами, описывающими компоненты модели (Y,U,V)). Присутствующие в формуле коэффициенты перевода постоянны и определяются особенностями человеческого восприятия. Для полутонового изображения важно только значение первой составляющей.

Оператор Собеля

В применяемом методе, использующем специальные ядра, известные как «операторы Собеля», действующие в области изображения размером 3*3 используется весовой коэффициент 2 для средних элементов. Коэффициенты ядра выбраны так, чтобы при его применении одновременно выполнялось сглаживание в одном направлении и вычисление пространственной производной – в другом. Маски, используемые оператором Собеля, отображены ниже:

Таблица 2. Маски, используемые оператором Собеля.

По ним получаем составляющие градиента G_x и G_y :

$$\begin{aligned}
 &) - (z_1 + 2z_2 + z_3) \\
 &) - (z_1 + 2z_4 + z_7) \quad (2.2)
 \end{aligned}$$

Для вычисления величины градиента эти составляющие необходимо использовать совместно:

≈

Результат ~~показывает насколько «резко»~~ или «главно» ~~меняется~~ яркость изображения в каждой точке, а значит, вероятность нахождения точки на грани, а также ориентацию границы. Результатом работы оператора Собеля в точке области постоянной яркости будет нулевой вектор, а в точке, лежащей на границе областей различной яркости — вектор, пересекающий границу в направлении увеличения яркости. Также,

результат работы — перевод фотографии в граничные контуры (это можно видеть на преобразованной картинке):



Рис. 2.5 Перевод изображения в граничные контуры.

Пороговая классификация

Далее, то же полутоновое изображение подвергается пороговой классификации, или выбору порога по яркости. Смысл такого порога заключается в том, чтобы отделить искомый светлый объект и темный фон, где объект — это совокупность тех пикселей, яркость которых превышает порог ($I > T$), а фон — совокупность остальных пикселей, яркость которых ниже порога ($I < T$). Примерный алгоритм работы с глобальным порогом в автоматическом режиме выглядит так:

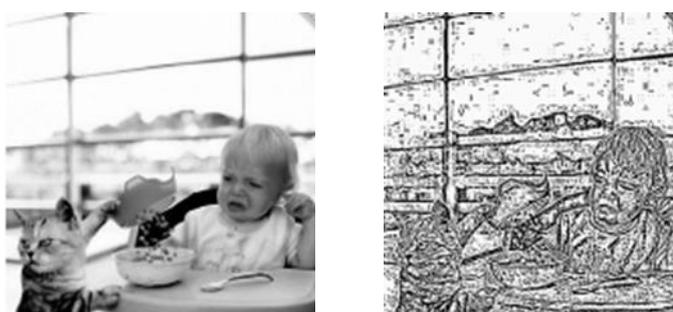


Рис. 2.6 Изображение подвергнутое пороговой классификации.

- Выбирается начальная оценка порога T — это может быть средний уровень яркости изображения (полусумма минимальной и максимальной яркости, $\min + \max / 2$) или любой другой критерий порога;

- Производится некая сегментация изображения с помощью порога T — в результате образуются две группы пикселей: G_1 , состоящая из пикселей с яркостью больше T , и G_2 , состоящая из пикселей с яркостью меньше или равной T ;
- Вычисляются средние значения μ_1 и μ_2 яркостей пикселей по областям G_1 и G_2 ;
- Вычисляется новое значение порога $T = 0.5 * (\mu_1 + \mu_2)$;
- Повторяются шаги со второго по четвертый до тех пор, пока разница значений порога T предыдущего и T вычисленного в последней итерации не окажется меньше наперед заданного параметра ε .

Бинаризация методом Отсу

Оператор Собеля не может этого сделать, так как дает очень много шумов и помех, поэтому полутоновое изображение необходимо бинаризовать. Значения яркостей пикселей изображения можно рассматривать как случайные величины, а их гистограмму — как оценку плотности распределения вероятностей. Если плотности распределения вероятностей известны, то можно определить оптимальный (в смысле минимума ошибки) порог для сегментации изображения на два класса c_0 и c_1 (объекты и фон). Суть метода Отсу заключается в том, чтобы выставить порог между классами таким образом, чтобы каждый из них был как можно более «плотным». Если выразиться математическим языком, то это сводится к минимизации внутриклассовой дисперсии, которая определяется как взвешенная сумма дисперсий двух классов (т.е. это сумма отклонений от математических ожиданий данных классов):

$$\sigma_w^2 = w_1 * \sigma_1^2 + w_2 * \sigma_2^2 \quad (2.4)$$

где σ_w — внутриклассовая дисперсия, σ_1 и σ_2 — дисперсии, а w_1 и w_2 — вероятности первого и второго классов соответственно.

Общая схема быстрого алгоритма такова:

- Вычисляем гистограмму (один проход через массив пикселей). Дальше нужна только гистограмма; проходов по всему изображению больше не требуется.

- Начиная с порога $t = 1$, проходим через всю гистограмму, на каждом шаге пересчитывая дисперсию $\sigma_b(t)$. Если на каком-то из шагов дисперсия оказалась больше максимума, то дисперсия обновляется и $T = t$.

- Искомый порог равен T . В более точной реализации есть параметры, которые позволяют убыстрить алгоритм, к примеру, проход через гистограмму делается не от 1 до 254, а от \min до $(\max - 1)$ яркости. «Классическая» бинаризация показана на рисунке 2.7.



Рис. 2.7 Классическая бинаризация изображения.

Детектор границ Канни

Итак, исходя из предыдущих шагов обработки изображения, получены края изображения и разбиение порогом в соответствии со значениями яркости. Теперь, чтобы закрепить результат, необходимо применить детектор границ Канни (Canny) для нахождения и окончательного связывания краёв объектов на изображении в контуры.

С помощью алгоритма Канни решаются такие задачи:

- Подавление «ложных» максимумов. Только некоторые из максимумов отмечаются как границы;

- Последующая двойная пороговая фильтрация. Потенциальные границы определяются порогами. Разбиение на тонкие края;

- Подавление всех неоднозначных краёв, не принадлежащих каким-либо границам областей. Связывание краёв в контуры. После подавления локальных неопределенностей, края становятся более точными и тонкими. Таким образом, получается двоичное изображение, содержащее границы (так называемые «тонкие края»).



Рис. 2.8 Двоичное изображение.

Выделение границ Канни использует два порога фильтрации:

- если значение пикселя выше верхней границы – он принимает максимальное значение и граница считается достоверной, пиксель выделен;
- если ниже – пиксель подавляется;
- точки со значением, попадающим в диапазон между порогами, принимают фиксированное среднее значение;
- затем найденные пиксели, со средним фиксированным добавляются к группе, если они соприкасаются с группой по одному из четырех направлений.

Причесывание изображения

После детектора Канни изображение подвергается морфологической операции дилатации – расширения, утолщения найденных границ за счет того, что по ним пробегается структурообразующее множество.



Рис. 2.9 Морфологическая операция дилатации – расширения.

Дополнительные примитивы Хаара

Итак, для повышения возможностей алгоритма и качества нахождения наклоненных черт лица предлагается такой вариант новых типов признаков, как на рисунке 2.10.

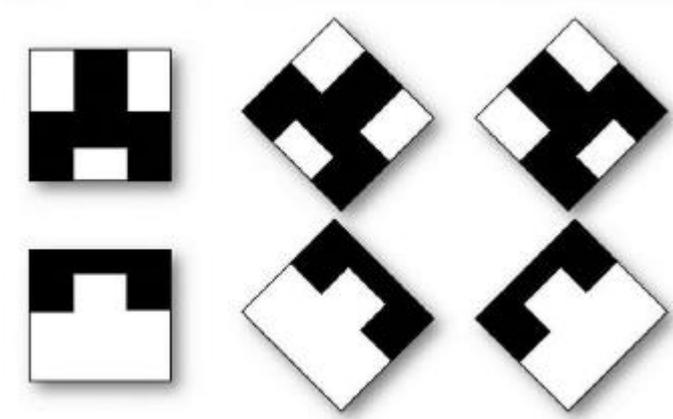


Рис. 2.10 Признаки Хаара.

Такие признаки были выбраны не случайно, а в связи с силуэтами искомого черт лица. С помощью данных признаков программа будет искать намного быстрее нос, брови и рот, но при этом на другие черты лица будет тратить чуть больше времени, чем в стандартном алгоритме. Однако дополнительные признаки стоит внедрять, так как в целом, выигрываются миллисекунды распознавания, и при этом увеличивается точность распознавания.

3. **Корректировка результатов бесконтактного сканирования и фотографий документов**

Хочу рассказать в своем опыте, как можно с минимальными затратами привести призмобразный, в лучшем случае, документ, который мы идентифицировали на снимке, к прямоугольной форме, желательно с сохранением исходных пропорций. Экзотические случаи, вроде пятиугольных или овальных документов мы пока не рассматриваем, хотя, вопрос интересный.

Дано:

1. снимок прямоугольного документа с искажениями
2. контуры документа на снимке

Задача: привести документ к исходной форме кратчайшим путем

Входные условия:

1. пропорции исходного документа нам точно не известны
2. расстояние до плоскости, на которой лежит документ нам не известно
3. референсных объектов, на которые можно ориентироваться (например, правильный квадрат, попавший в объектив) на снимке нет

Решение:

Итак, чтобы решить задачу в целом, предлагаем разбить её на две отдельные части:

1. Нахождение, собственно, искаженного контура документа на отсканированном изображении.
2. Определение правильных пропорций документа, в которые исходный искаженный контур должен быть отображен для того, чтобы получить выровненный документ.

Можно, конечно, было попытаться изобрести велосипед, и некоторым это до сих пор удается, но мы пошли более легким путем и использовали инструментарий OpenCV. Работаем мы по большей части в

среде .NET, через C# Wrapper OpenCVSharp. Также OpenCVSharp доступен в виде Nuget-пакета в среде Visual Studio. «Вот это всё» и будем использовать.

Рассмотрим основные интересные моменты в решении задачи по исправлению перспективного изображения на следующем изображении:

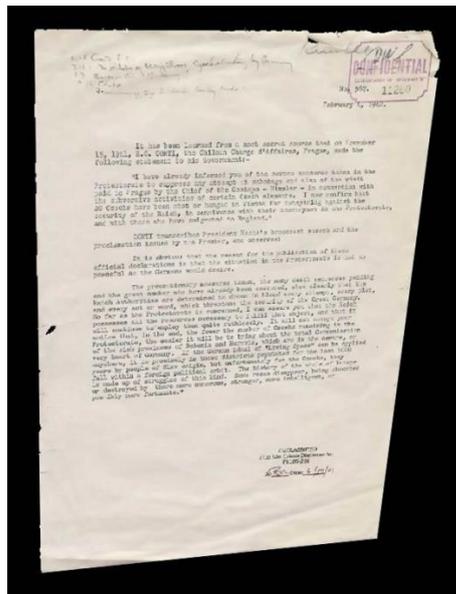


Рис. 2.11 Исправлению перспективного изображения.

Для того чтобы найти контур на представленном изображении, необходимо избавиться от мелких деталей, которые могут мешать. Это можно сделать, применив «заклинание размытия» по Гауссу малой мощности, предварительно сконвертировав изображение в оттенки серого:

```
imgSource.CvtColor(imgGrayscale, ColorConversion.BgrToGray);  
imgSource.Smooth(imgSource, SmoothType.Gaussian, 15);
```

Вот, что получилось в результате применения вышеописанной цепочки:

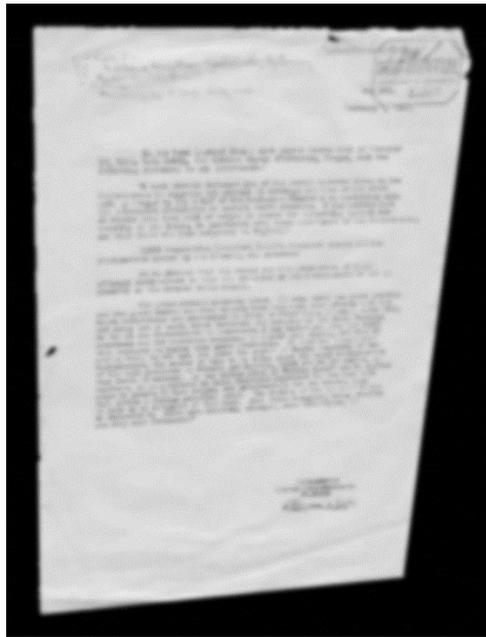


Рис. 2.12 Предварительная конвертация изображения в оттенки серого.

Далее необходимо сделать изображение черно-белым:

```
imgSource.Threshold(imgSource, 0, 255, ThresholdType.Binary |  
ThresholdType.Otsu);
```

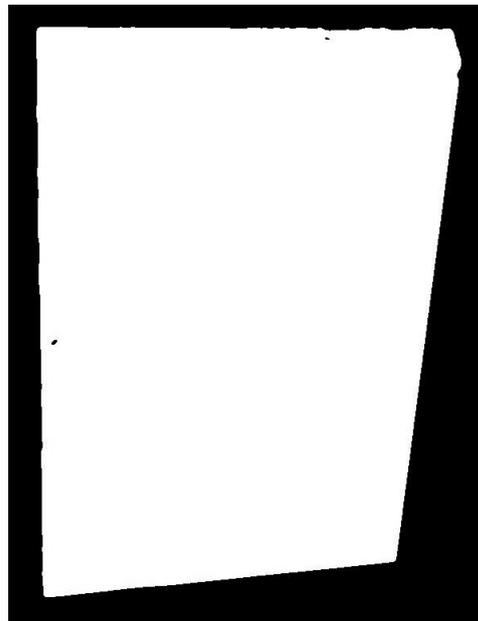


Рис. 2.13 Черно белое изображение.

На полученном изображении легко найти контур документа. Будем искать максимальный внешний контур. В OpenCVSharp есть замечательный класс CvContourScanner, который может перечислять все найденные контуры изображения. С использованием Linq можно эти контуры отсортировать по площади и взять первый, который и будет самым максимальным.

```
using (var storage = new CvMemStorage())
using (var scanner = new CvContourScanner(image, _storage,
CvContour.SizeOf, ContourRetrieval.External, ContourChain.ApproxSimple))
{
    var largestContour = scanner.OrderBy(contour =>
Math.Abs(contour.ContourArea())).FirstOrDefault();
```

Если нарисовать найденный контур, то получается следующее изображение:

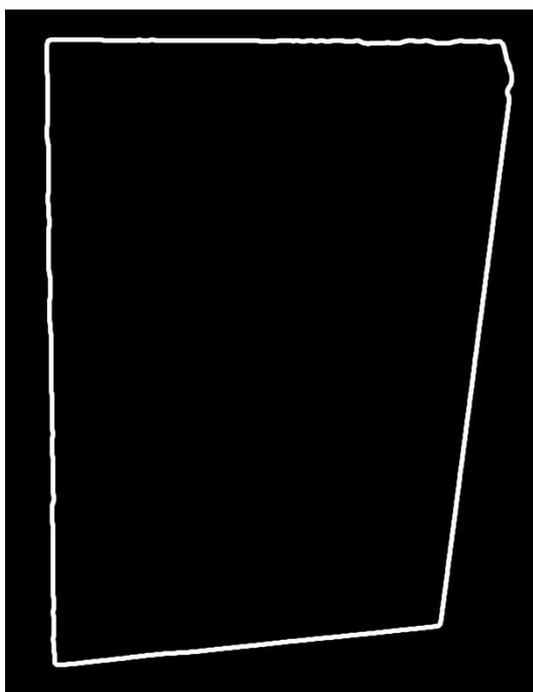


Рис. 2.14 Результат распознавания контура.

Нашли контур. Однако он мало что может показать – необходимо знать точно координаты всех угловых точек – точек пересечения сторон документа. Очевидно, что для нахождения координат этих точек желательно описать стороны найденного контура уравнениями прямой линии. В библиотеке OpenCV есть инструмент, использующий преобразование Хара. «Кастуем» этот метод на изображение, полученное на предыдущем шаге:

```
var lineSegments = imgSource.HoughLines2(storage,
HoughLinesMethod.Probabilistic, 1, Math.PI / 180.0, 70, 100,
1).ToArray<CvLineSegmentPoint>();
```

Но этот метод не вернет нам 4 линии, которые мы бы ожидали получить. Их будет 100, а может быть 200, а может вообще не быть. Дело в том, что данный метод ищет все участки, которые были приняты за линии, и удовлетворяющие входным параметрам. Тем не менее, с этими данными уже можно что-то делать, например, разложить их по кучкам: вертикальные отдельно, горизонтальные отдельно:

```
var verticalSegments = segments .Where(s => Math.Abs(s.P1.X - s.P2.X)
< Math.Abs(s.P1.Y - s.P2.Y)) .ToArray(); var horizontalSegments =
segments .Where(s => Math.Abs(s.P1.X - s.P2.X) >= Math.Abs(s.P1.Y -
s.P2.Y)) .ToArray();
```

Отрезки линий, которые «динамичнее» изменяются по вертикали – это вертикальные; по горизонтали – горизонтальные. Стало намного проще, можно даже нарисовать, что получилось:

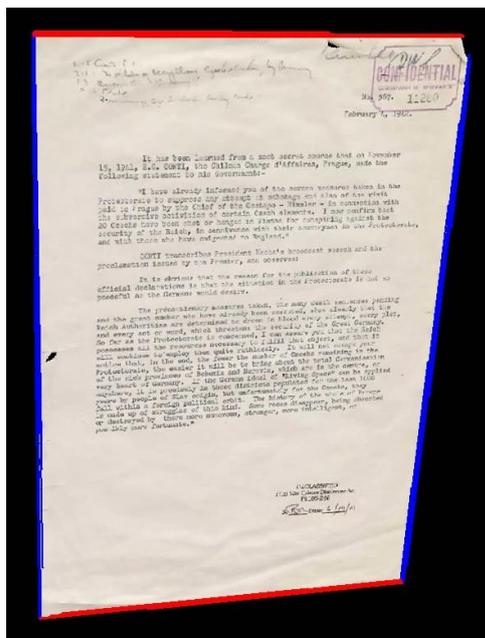


Рис. 2.15 Результат

Далее, попробуем найти точки пересечения всех вертикальных и горизонтальных линий. Смотрим, что получается:

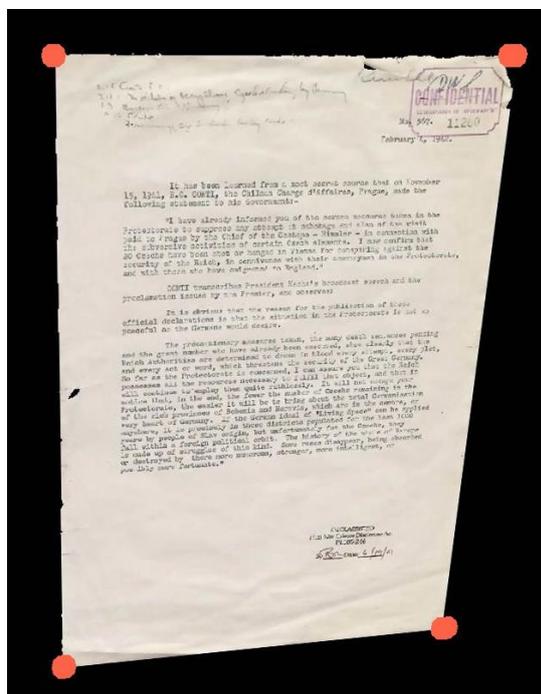


Рис. 2.16 Результат.

Осталось теперь отсортировать все найденные точки по часовой стрелке относительно центра масс этих точек:

– среднее арифметическое по каждой из координат). После этого из отсортированного массива создаем контур и аппроксимируем его средствами OpenCVSharp:

```
contour = contour.ApproxPoly(CvContour.SizeOf, storage, ApproxPolyMethod.DP, contour.ArcLength() * 0.02, true);
```

Мы, наконец-то получили искомые точки искаженного контура:

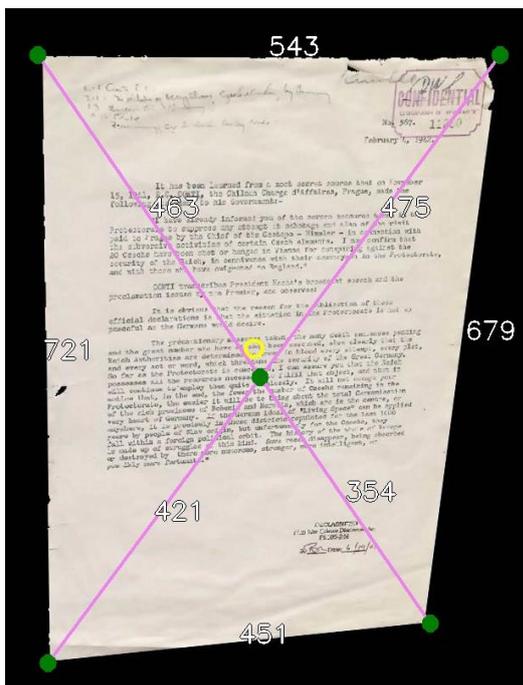


Рис. 2.17 Результат.

Единственное, что осталось сделать – это вычислить угловые точки выровненного контура с тем, чтобы потом отобразить в них искаженные точки. Если быть точным, необходимо найти пропорции документа, которые восприятие человеческого глаза могло посчитать верными. Основная проблема, которая встала перед нами – это отсутствие каких-либо начальных данных, по которым можно было бы вычислить правильные пропорции документа. Не было информации ни о том, под каким углом был отсканирован документ, ни о фокусном расстоянии.

Решение, описанное далее, не является универсальным для всех случаев перспективного искажения и не дает 100% точности восстановления исходных пропорций документа. Однако, для наших целей и с нашими вводными, это решение компактно, вполне жизнеспособно, не лишено элегантности, и дает неплохие результаты.

Мы решили пойти простым путем: взять максимальные по длине горизонтальную и вертикальную стороны искаженного контура и использовать эти величины в качестве размеров выровненного контура. Однако этот метод давал приемлемые результаты лишь на небольших искажениях. Более серьезные искажения, такие как это, например:

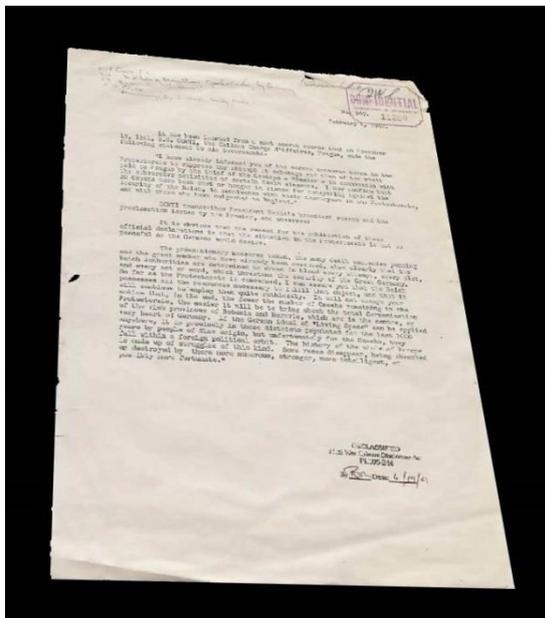


Рис. 2.18 Результат.

приводили к получению подобных результатов:

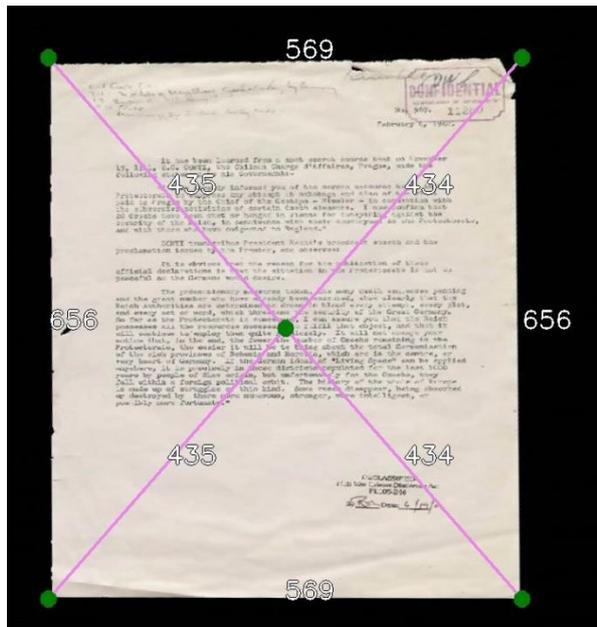


Рис. 2.19 Результат.

Квадратный документ нам не нужен и поэтому необходимо было придумать что-то более качественное. Опытным путем было замечено, что на искаженных документах наблюдается отклонение центра масс угловых точек контура от точки пересечения диагоналей контура (рисунок 10, желтое кольцо – центр масс, зеленый круг – точка пересечения диагоналей):

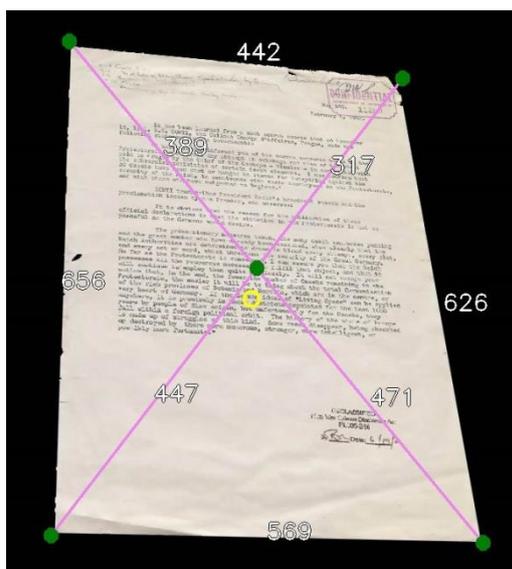


Рис. 2.20 Результат.

Нетрудно догадаться, что на «ровных» документах эти точки совпадают. Если же есть какое-то искажение, то обязательно будет наблюдаться отклонение и чем искажение больше, тем больше и отклонение. Вооружившись этим фактом и еще чуть-чуть поисследовав, мы пришли к простой формуле, точнее к двум:

где:

- это отклонения центра масс от точки пересечения диагоналей, соответственно;
- размеры результирующего контура;
- размеры искаженного контура.

А вот результат применения этой формулы:

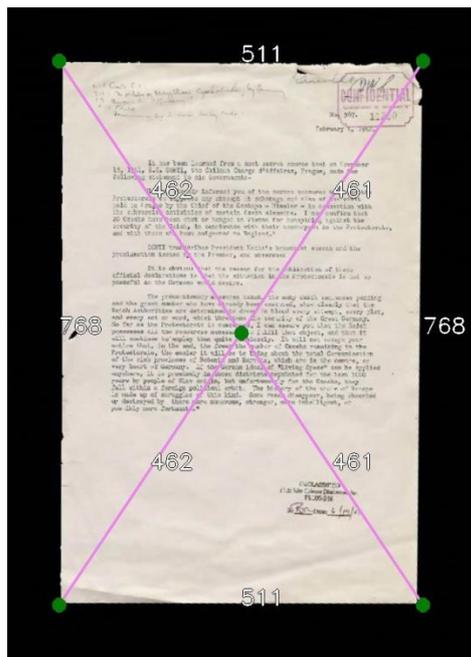


Рис. 2.21 Результат.

Для сравнения приведем пропорции исходного документа, отсканированного без искажений:

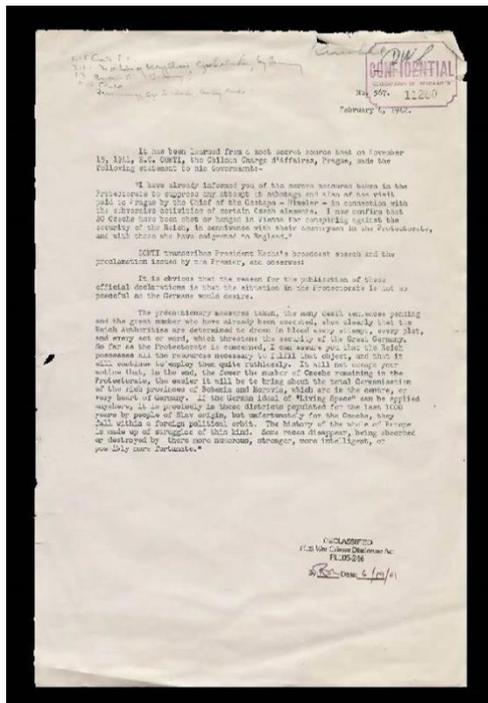


Рис. 2.22 Результат.

Выводы по главе II

Данный подход универсален, расширяем и если обучено достаточное количество каскадов классификаторов, то метод нахождения работает очень быстро и практически безошибочно. Подробно рассмотрен механизм работы алгоритма Виолы-Джонса (Viola-Jones) и описаны предложенные подходы к повышению эффективности решения задачи обнаружения эмоций на лице человека.

Глава III. Разработка прикладного программного обеспечения распознавания

1. Роль и значения программного обеспечения для решения задачи распознавания текстов и лиц

В данной главе рассматривается пакет прикладных программ, предназначенный для решения задач распознавания текстов с изображений и лиц с видеопотока. Алгоритмическую базу пакета прикладных программ составляют процедуры, рассмотренные выше (глав II).

Приводятся результаты экспериментальных исследований по оценке сходимости алгоритмов, предложенных в данной работе, а также результаты решения практической задачи.

Разработки программных систем анализа данных и прогноза по прецедентам активно ведутся в Узбекистане, а также в ведущих зарубежных странах. Прежде всего это статистические пакеты обработки данных и визуализации, в основе которых лежат методы различных разделов математической статистики, проверка статистических гипотез, регрессионный анализ, дисперсионный анализ, анализ временных рядов и др. Использование статистических программных продуктов стало стандартным и эффективным инструментом анализа данных, и, прежде всего, начального этапа исследований, когда находятся значения различных усредненных показателей, проверяется статистическая достоверность различных гипотез, находятся регрессионные зависимости.

Вместе с тем статистические подходы имеют и существенные недостатки. Они позволяют оценить (при выполнении некоторых условий) статистическую достоверность значения прогнозируемого параметра, гипотезы или зависимости, однако сами методы вычисления прогнозируемых величин, выдвижения гипотез или нахождения зависимостей имеют очевидные ограничения. Прежде всего, находятся

усредненные по выборке величины, что может быть достаточно грубым представлением об анализируемых или прогнозируемых параметрах. Любая статистическая модель использует понятия «случайных событий», «функций распределения случайных величин» и т.п., в то время как взаимосвязи между различными параметрами исследуемых объектов, ситуаций или явлений являются детерминированными. Само применение статистических методов подразумевает наличие определенного числа наблюдений для обоснованности конечного результата, в то время как данное число может быть существенно больше имеющегося или возможного. Т.е. в ситуациях анализа в принципе непредставительных данных или на этапах начала накопления данных, статистические подходы становятся неэффективными как средство анализа и прогноза.

В последние годы появились узкоспециализированные пакеты интеллектуального анализа данных. Для данных пакетов часто характерна ориентация на узкий круг практических задач, а их алгоритмической основой является какая-либо одна из альтернативных моделей, использующая нейронную сеть, решающие деревья, ограниченный перебор, и т.п. Ясно, что подобные разработки существенно ограничены при практическом использовании. Во-первых, заложенные в них подходы не являются универсальными относительно размерностей задач, типа, сложности и структурированности данных, величины шума, противоречивости данных, и т.п. Во-вторых, созданные и «настроенные» на решение определенных задач, они могут оказаться совершенно бесполезными для других. Наконец, множество задач, представляющих интерес практическому пользователю, обычно шире возможностей отдельного подхода. Например, пользователю может быть важно, иметь численную характеристику надежности некоторого прогноза, но «решающее дерево» ее не вычисляет. «Нейронная сеть» выступает в роли «черного ящика», предлагающего некоторый прогноз без его обоснования. Логические методы распознавания позволяют выявлять логические

закономерности в данных и использовать их при прогнозировании, но при наличии линейных зависимостей между признаками и прогнозируемой величиной точность прогноза, сделанного «линейной машиной», может быть заметно выше.

Таким образом, на настоящем уровне развития методов решения задач, анализа данных и распознавания, представляется предпочтительным путь создания программных средств, включающих основные существующие разнообразные подходы. В данном случае повышаются шансы подбора из имеющихся алгоритмов такого алгоритма, который обеспечит наиболее точное решение интересующих пользователя задач на новых данных. Другим важным атрибутом систем анализа и классификации должно быть наличие средств автоматического решения задач распознавания и классификации коллективами алгоритмов. Действительно, стандартной ситуацией является наличие нескольких альтернативных алгоритмов или решений, равнозначных для пользователя. Для выбора из них одного наиболее предпочтительного не хватает информации. Тогда естественной альтернативой выбору является создание на базе имеющихся алгоритмов или решений новых, более предпочтительных.

В алгебраическом подходе новые алгоритмы распознавания строятся в виде полиномов над исходными алгоритмами (применение алгебраических корректоров) или в виде специальных булевских функций (логических корректоров). Теоретическим базисом является теорема о существовании для произвольного алгоритма распознавания ему эквивалентного стандартного алгоритма, представимого в виде произведения распознающего оператора и решающего правила. Это позволяет описать основные результаты вычислений произвольных алгоритмов распознавания в стандартном виде с помощью числовых матриц оценок («мер принадлежности» объектов к классам) и информационных матриц окончательных ответов (классификаций).

Матрицы оценок различных распознающих алгоритмов являются «исходным материалом» для синтеза в виде полиномов новых матриц оценок, которые задают основу нового скорректированного решения задачи распознавания. Алгебраический подход позволяет строить алгоритмы, безошибочные на «обучающем» материале или совершающие меньшее число ошибок, чем каждый из исходных алгоритмов.

В настоящее время существует множество разнообразных подходов и конкретных эвристических алгоритмов для решения задач кластерного анализа (таксономии, или классификации без учителя), когда требуется найти естественные группировки похожих объектов (кластеры) по заданной выборке их векторных признаков описаний. Решения, найденные различными алгоритмами, могут существенно отличаться друг от друга и даже фактически не соответствовать заложенной в данных действительности. Поиск наилучшего решения затруднен отсутствием общепризнанных универсальных критериев качества решений. Методы построения оптимальных коллективных решений в задачах кластерного анализа позволяют находить такие группировки объектов, которые являются эквивалентными с позиций сразу нескольких исходных алгоритмов. Оптимальные кластеризации находятся в результате решения специальных дискретных оптимизационных задач на перестановках.

2. Назначения, состав и режим работы ППО «DPro»

Целью разработанного прикладного программного обеспечения является распознавание текстов с изображений и лиц (образов) с видеопотоков.

Для разработки данного программного обеспечения выбран алгоритмический язык программирования C#.

Разработанное программное обеспечение состоит из 9 компонентов, все из которых по выполняемым функциям объединены в три модуля.

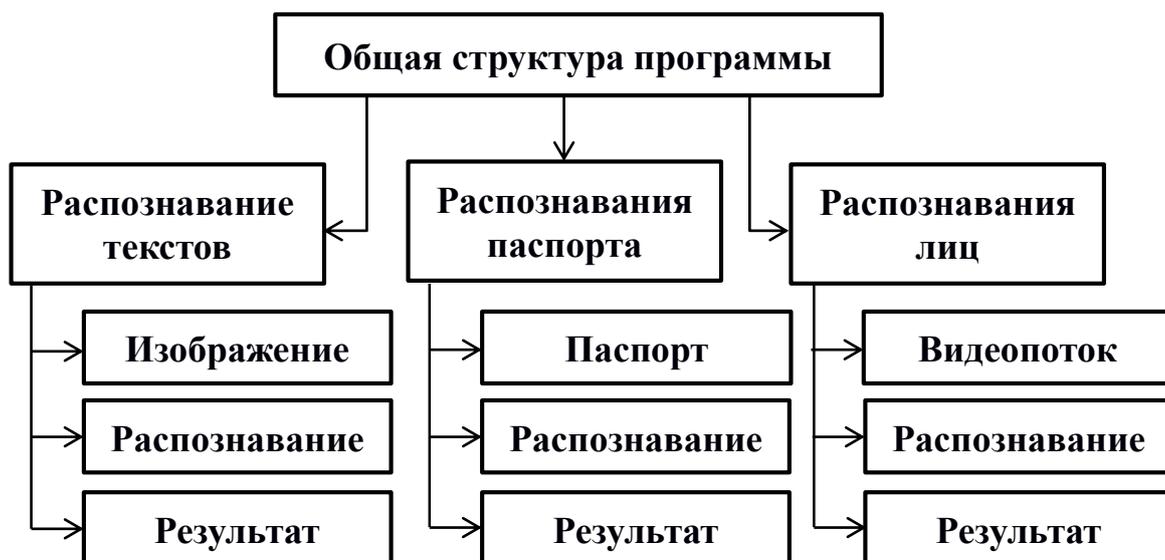


Рис. 3.1 Общая структура программы.

Основу рабочей среды программы составляет стандартный графический интерфейс Windows. После запуска программы на экран выводится следующее окно:



Рис. 3.2 Окно запуска программы.

Потом в трейбаре появляется иконка. Здесь мы можем запустить один из трех модулей («Распознавание текстов», «Распознавания паспорта» и «Распознавание лиц»).

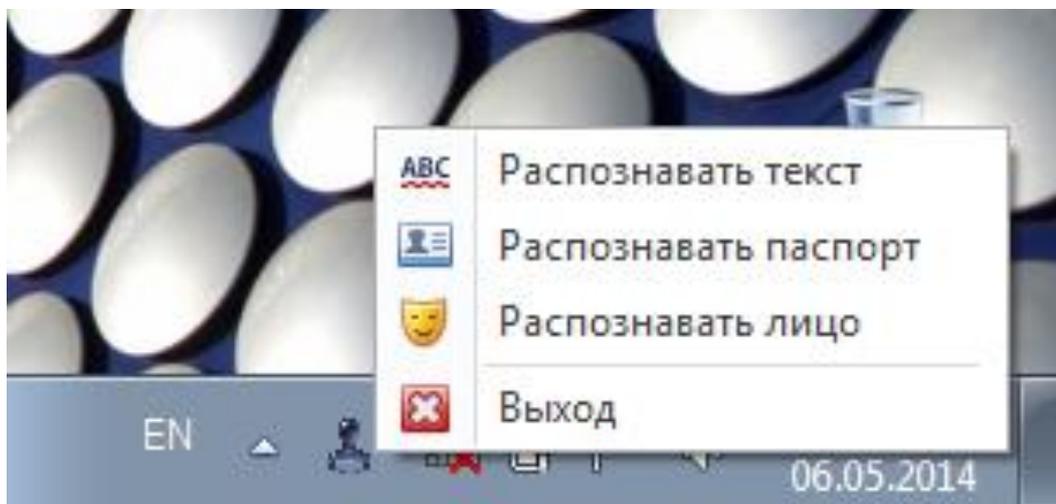


Рис. 3.3 Выбор модулей программы из трейбара

3. Описания модуля «Распознавание текстов»

Модуль «Распознавание текстов» предназначен для распознавания текстов с изображений. Модуль работает в двойном режиме: автоматический и ручной режим. Автоматический режим запускается автоматически после загрузки изображения.

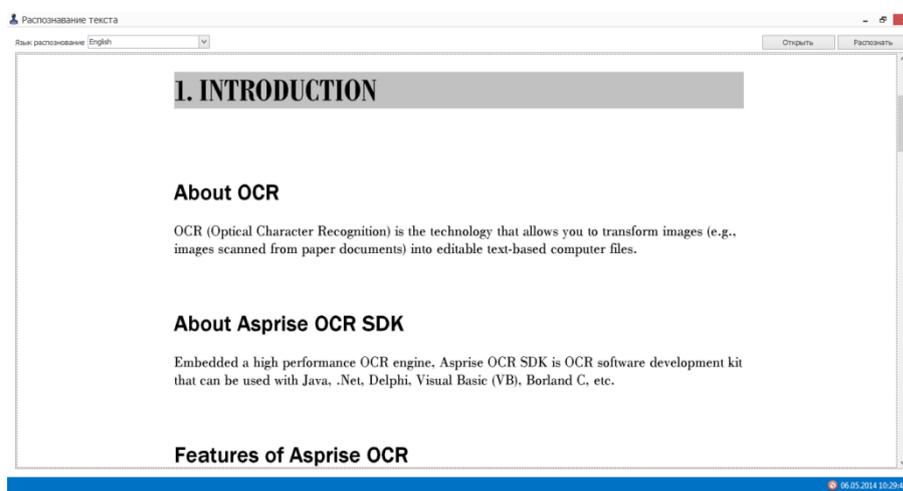


Рис 3.4 Загрузка изображения в программу.

Если пользователь хочет изменить язык распознавания, но не хочет заново загрузить изображения, чтобы распознавать его, то пользователь должен просто нажать на кнопку «Распознать» на верхнем правом углу. Это второй, ручной режим работы программы.

После распознавания откроется второй экран с результатом:

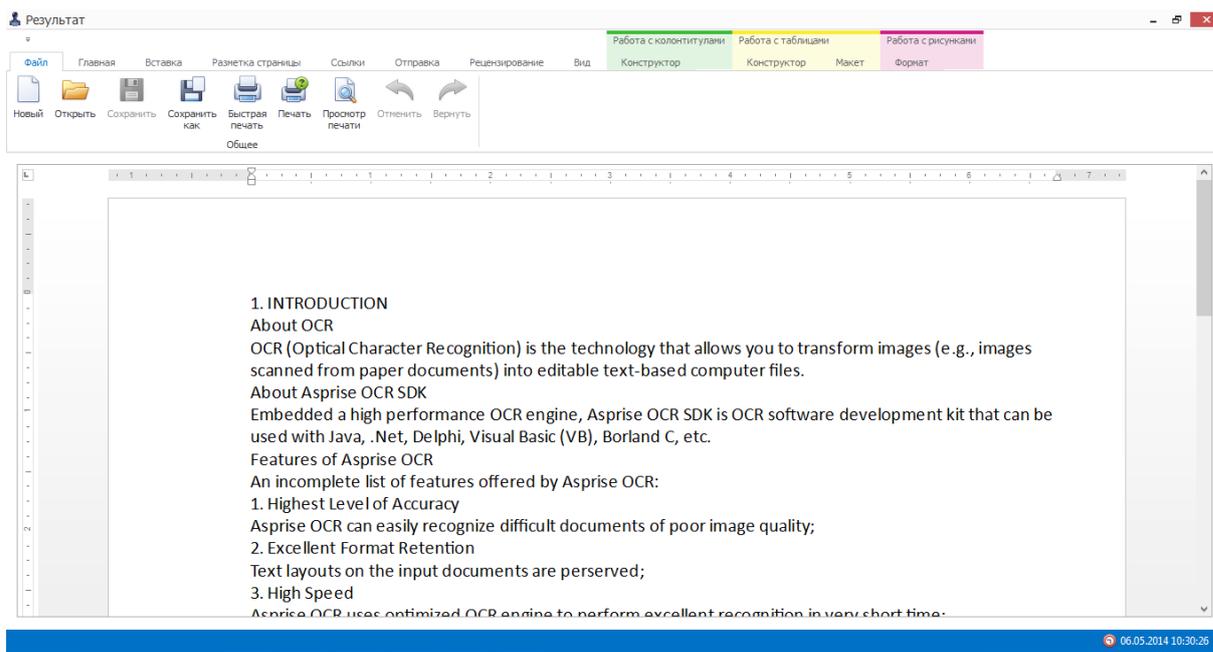


Рис. 3.5 Окно с результатом модуля «Распознавание текстов»

С помощью этого окна пользователь может отредактировать полученный результат, сохранить, распечатать, отправить по почте и т.д. на свой вкус и желания.

4. Описания модуля «Распознавания паспорта»

Модуль «Распознавания паспорта» предназначен для распознавания простых и биометрических паспортов, полученного со сканера или из жесткого диска. Модуль работает в автоматическом и в ручном режиме. Автоматический режим запускается автоматически после сканирования или загрузки изображения паспорта.

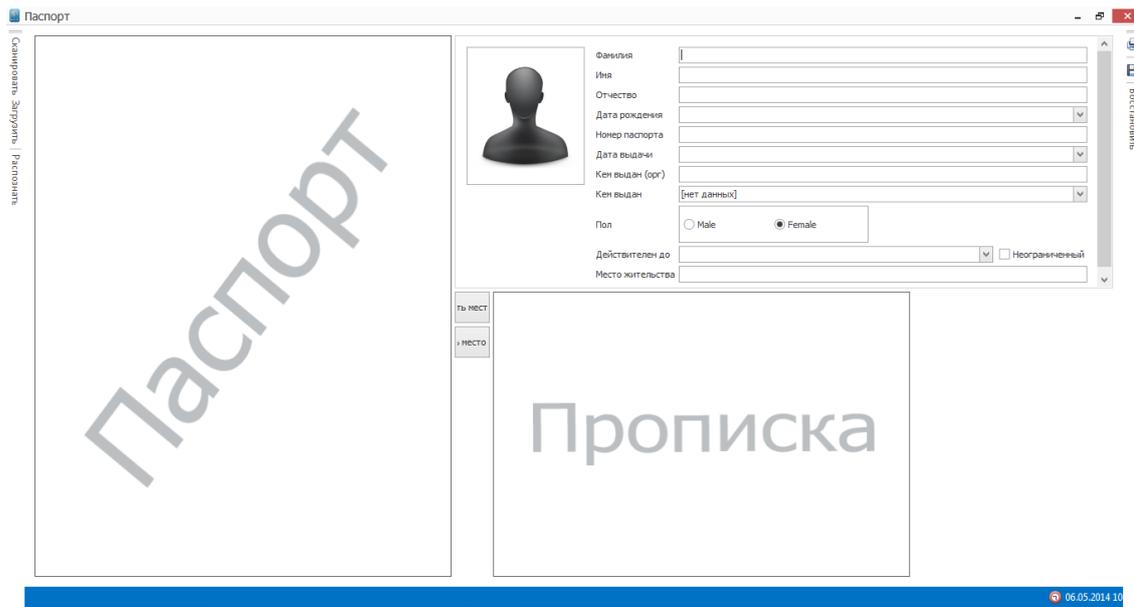


Рис. 3.6 Главное окно модуля «Распознавания паспорта».

Чтобы отсканировать паспорт и получить изображения, мы должны нажать на кнопку «Сканировать» в верхнем левом углу, чтобы загрузить изображение паспорта с жесткого диска мы должны нажать на кнопку «Загрузить» в верхнем левом углу. После сканирования (загрузки) паспорта программа распознает его и выводит на экран результат:

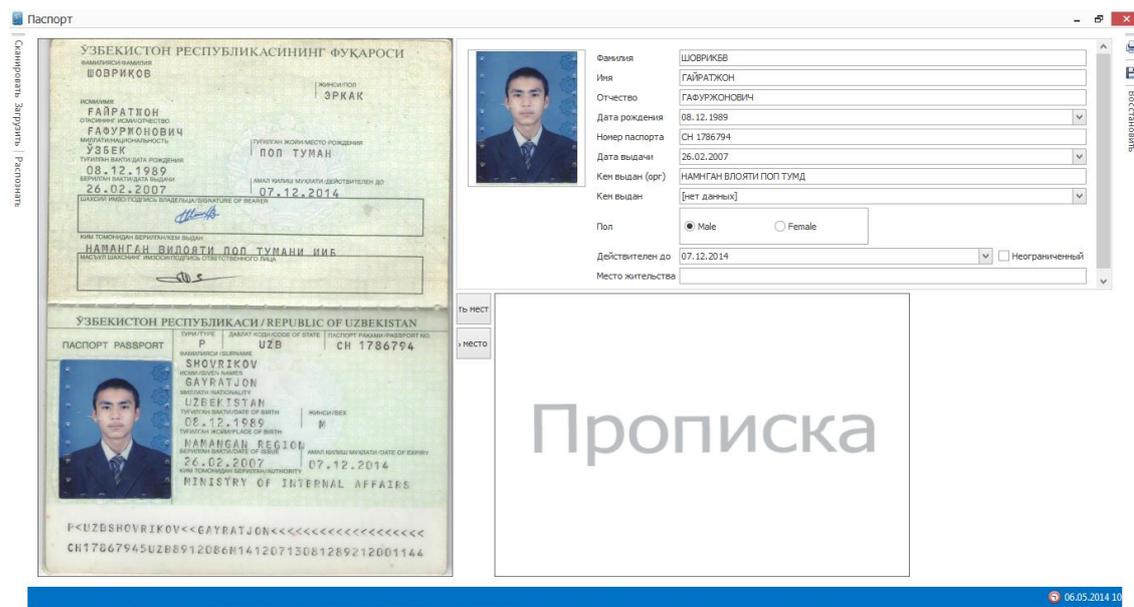


Рис. 3.7 Результат модуля «Распознавания паспорта».

На качество результата влияет качество и расположение изображения.

Этот модуль можно использовать в биллинговых системах или учете клиентов компании.

5. Описание модуля «Распознавания лиц»

После запуска модуля открывается следующее окно:

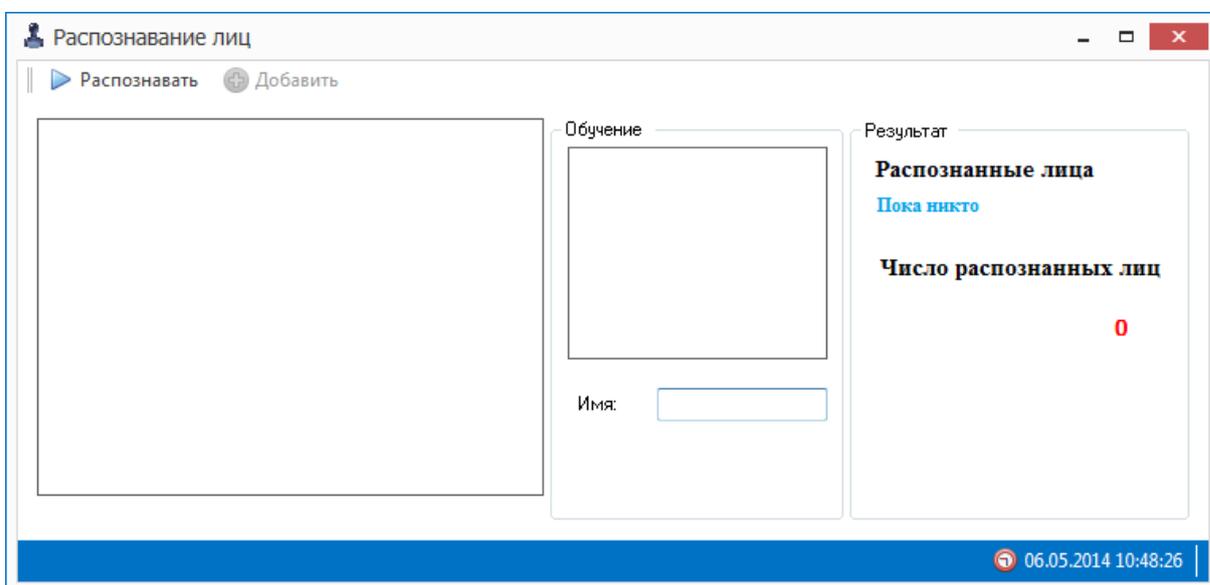


Рис. 3.8 Главное окно модуля «Распознавания лиц».

Для начала работы нам необходимо нажать на кнопку «Распознавать». После чего программа начинает отображать кадры, получаемые с веб камеры. В следующем рисунке мы увидим, как программа распознаёт и выделяет квадратом область лица в кадре, полученной с веб камеры, но не идентифицирует его.

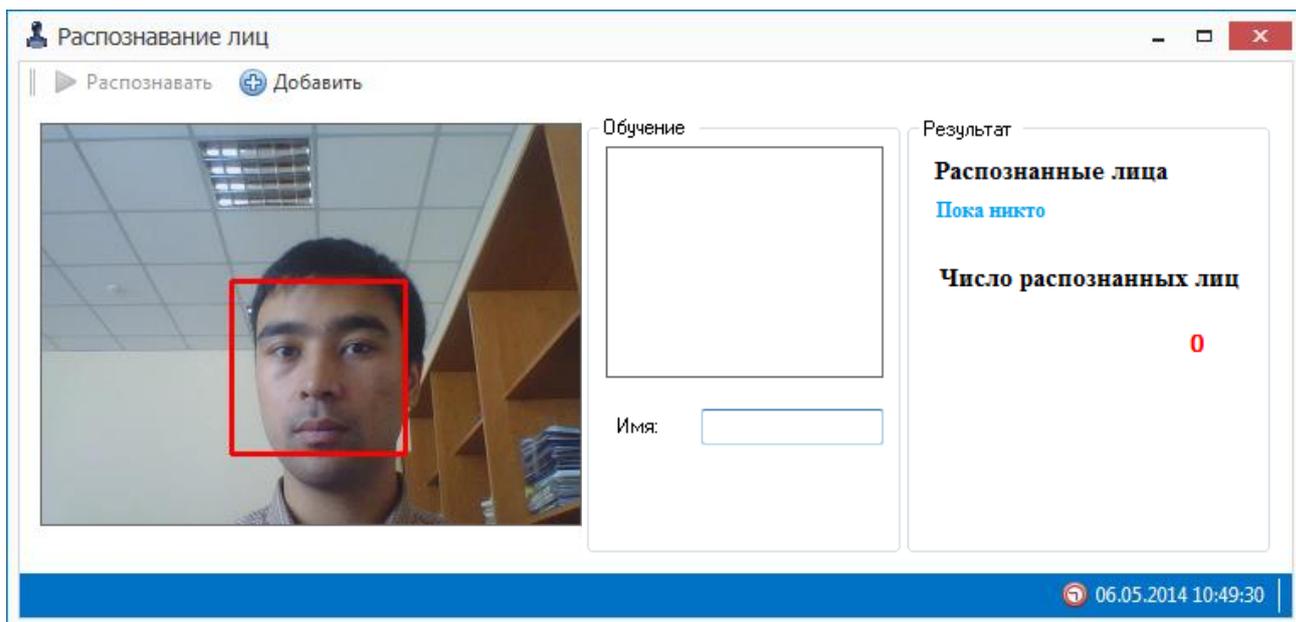


Рис. 3.9 Окно распознавания модуля «Распознавания лиц».

Для идентификации лица заполняем поле «Имя» куда вписываем имя идентифицируемого человека и нажимаем кнопку «Добавить». При этом выделенная область лица запоминается программой как идентификатор. Этот процесс называется обучением системы, и мы можем увидеть этот результат в следующем рисунке.

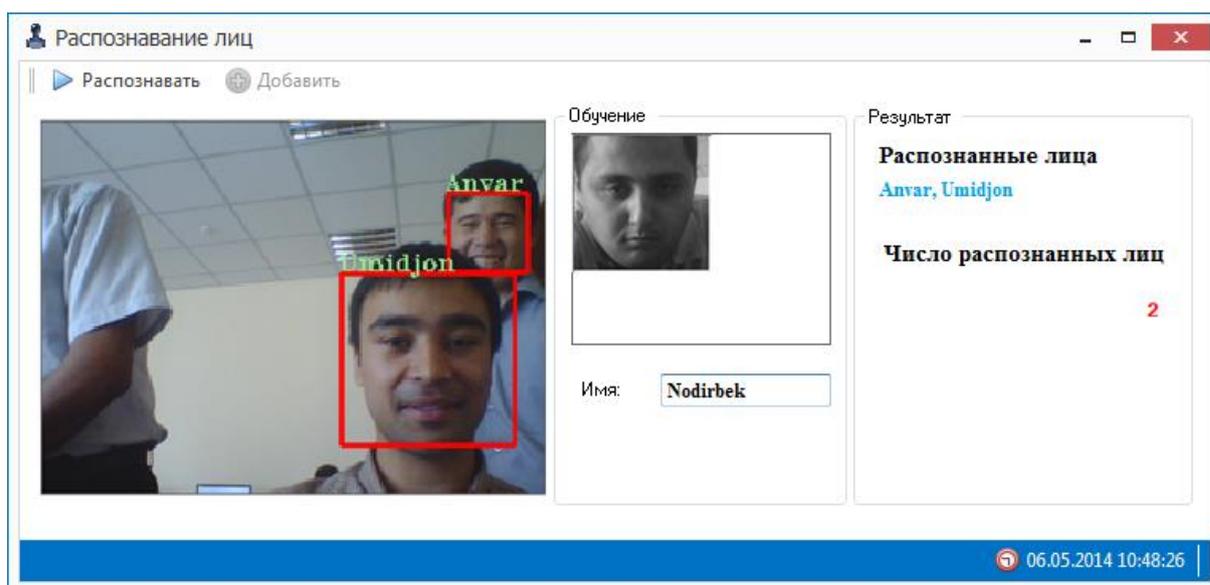


Рис. 3.10 Распознавание и идентификация лиц в реальном времени

Как мы видим, программа делает надпись над выделенной областью и записывает там имя идентифицируемого лица. Если в одно время в зону видимости программы попадают несколько идентифицированных лиц, то программа показывает имена этих лиц, а также количество идентифицированных лиц.

Выводы по главе III

1. Рассмотрен процесс работы программного обеспечения;
2. Получены ожидаемые результаты;
Выполнены процессы распознавания текста из изображения, распознавание паспорта из изображения и распознавание лица из видеопотока.

Заключение

Способность восприятия внешнего мира в форме образов и текста позволяет с определенной достоверностью узнавать бесконечное число объектов на основании ознакомления с конечным их числом, а объективный характер основного свойства образов и текста позволяет моделировать процесс их распознавания.

Полученные результаты данной работы:

- изучены алгоритмы распознавания текстов и собраны необходимые теоретические минимумы из существующих литератур;
- изучен алгоритм Виолы-Джонса;
- выбрана среда разработки и реализации метода разработки;
- сделаны обоснованные варианты и предложения об улучшении метода или работы программы;

Разработан программный комплекс для распознавания текстов, паспортов и лиц.

Перечень использованной литературы

I. Законы Республики Узбекистан

1. Закон Республики Узбекистан «Об информатизации»// 11 декабря 2003 года.
2. Закон Республики Узбекистан № 611-II от 29.04.2004 «Об электронном документообороте».

II. Книга Президента Республики Узбекистан Каримова И.А.

3. Каримов И.А. По пути преодоления последствий мирового кризиса, модернизации страны и достижения уровня развитых государств Ислам Абдуганиевич Каримов. Ред. Х. Кутлуков, // Ташкент: Узбекистан, 2010 – 248с.

Литературы

4. Чубукова И. А. Курс лекций по Data Mining — М., 2008. — 7 с.
5. Шоломов Д.Л. Синтаксические методы контекстной обработки в задачах распознавания текста. — М., 2007.
6. Арлазаров В. В., Постников В. В., Шоломов Д.Л. Cognitive Forms — система массового ввода структурированных документов. // "Управление информационными потоками" Сборник трудов Института системного анализа РАН./ М., УРСС, 2011 г.
7. Kevin P. Murphy. “Machine learning: a probabilistic perspective”.
8. Christopher M. Bishop. “Pattern recognition and machine learning”. October 1, 2010 – p 103, p 142.
9. Арлазаров В.В. Структурирование визуальных представлений информационной среды и методы определения надежности распознавания. — М., 2010. — 159 с, 168 с

10. Горелик А. Л., Скрипкин В. А. Методы распознавания. — 4-е изд. — М.: Высшая школа, 2010. — 262 с.

11. Фомин Я. А. Распознавание образов: теория и применения. — 2-е изд. — М.: ФАЗИС, 2012. — 429 с.

IV. Интернет сайты

HYPERLINK "http://ru.wikipedia.org/wiki/Теория_распознавания_образов"

--

--

--

--

--

--

--

--

--

--

--

open-cv.com

mgu-cv.com

-

E

R

L

I

N

K

"

h

t

Приложение

Программная реализация распознавание текста, паспорта и лица человека.

a) Реализация «Медианного фильтра».

```
public static void Median(ref Bitmap image)
{
    var arrR = new int[8];
    var arrG = new int[8];
    var arrB = new int[8];
    var outImage = new Bitmap(image);

    for (int i = 1; i < image.Width - 1; i++)
        for (int j = 1; j < image.Height - 1; j++)
            {
                for (int i1 = 0; i1 < 2; i1++)
                    for (int j1 = 0; j1 < 2; j1++)
                        {
                            var p = image.GetPixel(i + i1 - 1, j +
j1 - 1);

                                arrR[i1 * 3 + j1] = ((p.R + p.G + p.B) /
3) & 0xff;
                                arrG[i1 * 3 + j1] = ((p.R + p.G + p.B) /
3) >> 8 & 0xff;
                                arrB[i1 * 3 + j1] = ((p.R + p.G + p.B) /
3) >> 16 & 0xff;
                        }
                    Array.Sort(arrR);
                    Array.Sort(arrG);
                    Array.Sort(arrB);

                outImage.SetPixel(i, j, Color.FromArgb(arrR[3],
arrG[4], arrB[5]));
            }

    image = outImage;
}
```

Реализация «Монохромного фильтра».

```
public static void Monochrome(ref Bitmap image, int level)
{
    for (int j = 0; j < image.Height; j++)
    {
        for (int i = 0; i < image.Width; i++)
        {
            var color = image.GetPixel(i, j);
            int sr = (color.R + color.G + color.B) / 3;
            image.SetPixel(i, j, (sr < level ? Color.Black :
Color.White));
        }
    }
}
```

в) Распознавание текста.

```
public static class RecognizeHelper
{
    public static string ReadTextFromImage(string imagePath,
MODI.MiLANGUAGES language)
    {
        try
        {
            MODI.Document ModiObj = new MODI.Document();
            ModiObj.Create(imagePath);
            ModiObj.OCR(language, true, true);

            MODI.Image ModiImageObj =
(MODI.Image)ModiObj.Images[0];
            string text = ModiImageObj.Layout.Text;

            ModiObj.Close();

            return text;
        }
        catch (Exception ex)
        {
            throw new Exception(ex.Message);
        }
    }
}
```

```
public partial class RecognizeForm : WaitForm
{
    string filePath = string.Empty, rText = string.Empty;
    bool isClosed = false;
    MODI.MiLANGUAGES language = MODI.MiLANGUAGES.miLANG_ENGLISH;
```

```

        public RecognizeForm(string filePath, MODI.MiLANGUAGES
language)
    {
        InitializeComponent();
        this.progressBar1.AutoHeight = true;
        this.filePath = filePath;
        this.language = language;
    }

    public string RecognizedText
    {
        get { return rText; }
    }
    public bool IsClosed
    { get { return isClosed; } }

    #region Overrides

    public override void SetCaption(string caption)
    {
        base.SetCaption(caption);
        this.progressBar1.Caption = caption;
    }
    public override void SetDescription(string description)
    {
        base.SetDescription(description);
        this.progressBar1.Description = description;
    }
    public override void ProcessCommand(Enum cmd, object arg)
    {
        base.ProcessCommand(cmd, arg);
    }

    #endregion

    public enum WaitFormCommand
    {
    }

    private void RecognizeForm_Shown(object sender, EventArgs e)
    {
        rText = RecognizeHelper.ReadTextFromImage(filePath,
language);
        isClosed = true;
        DialogResult = System.Windows.Forms.DialogResult.OK;
    }
}

public partial class ResultForm : MainFormEx
{

```

```

string result = string.Empty;
public ResultForm(string result)
{
    InitializeComponent();
    this.result = result;
}

private void ResultForm_Shown(object sender, EventArgs e)
{
    richEditControl1.Text = result;
}
}

```

```

public partial class AccessPoint : SplashScreen
{
    private PassportData passportData = null;
    private List<string> listOfAuthority = null;

    public PassportData PassportData
    {
        get
        {
            return passportData;
        }
        set
        {
            passportData = value;
        }
    }
    public List<string> ListOfAuthority
    {
        get
        {
            return listOfAuthority;
        }
        set
        {
            listOfAuthority = value;
        }
    }
    public AccessPoint()
    {
        InitializeComponent();
    }

    #region Overrides

    public override void ProcessCommand(Enum cmd, object arg)
    {
        base.ProcessCommand(cmd, arg);
    }
}

```

```

    }

    #endregion

    public enum SplashScreenCommand
    {
    }

    private void exitButton_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        this.Close();
    }

    private void notifyIcon1_MouseDown(object sender,
MouseEventEventArgs e)
    {
        if (e.Button == System.Windows.Forms.MouseButtons.Right)
            popupMenu1.ShowPopup(Cursor.Position);
    }

    private void textButton_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        var textForm = new MainForm();
        textForm.Show();
    }

    private void passportButton_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        ScanPassport passportForm = new
ScanPassport(passportData, listOfAuthority);
        passportForm.Show();
    }

    private void faceButton_ItemClick(object sender,
DevExpress.XtraBars.ItemClickEventArgs e)
    {
        MultiFaceRecognizeForm frForm = new
MultiFaceRecognizeForm();
        frForm.Show();
    }

    private void AccessPoint_Shown(object sender, EventArgs e)
    {
        BackgroundWorker bw = new BackgroundWorker();
        bw.DoWork += (a, c) =>
        {
            DataBaseHelper manager = new DataBaseHelper();
            c.Result = manager.CreateDataBase();
        };
        bw.RunWorkerCompleted += (a, c) =>

```

```

    {
        var result = c.Result as ActionResult;
        if (result.Error != null)
        {
            XtraMessageBox.Show(result.Error.Message);
            this.Close();
        }
        else
        {
            notifyIcon1.Visible = true;
            this.Hide();
        }
    };
    bw.RunWorkerAsync();
}
}

```

2) Распознавание паспортов.

```

public static class EditImage
{
    public static Image CropImage(this Image image, Rectangle
selection)
    {
        try
        {
            Bitmap bmp = image as Bitmap;
            if (bmp != null)
            {
                Bitmap cropBmp = bmp.Clone(selection,
bmp.PixelFormat);
                image.Dispose();
                return cropBmp;
            }
            else
                return null;
        }
        catch
        {
            return null;
        }
    }

    public static bool ResizeImage(string imageLocation, Size
size)
    {
        try
        {
            Image img;

```

```

        using (Image imgToResize =
Image.FromFile(imageLocation))
        {
            img = (Image)(new Bitmap(imgToResize, size));
        }
        img.Save(imageLocation,
System.Drawing.Imaging.ImageFormat.Jpeg);

        return true;
    }
    catch
    { return false; }
}

public static Image ResizeImage(Image image, Size size)
{
    try
    {
        Image img = (Image)(new Bitmap(image, size));
        return img;
    }
    catch
    { return null; }
}
}

public partial class PleaseWaitForm : WaitForm
{
    string imageLocation = string.Empty;
    string lastName = string.Empty, firstName = string.Empty,
middleName = string.Empty;
    string passportNo = string.Empty, authority = string.Empty;
    DateTime birthday, dateOfIssue, dateOfExpire;
    bool sex, isUnlimited, isClosed = false, isBioPassport =
false;
    Image picture = null;

    private List<Dictionary<string, Rectangle>> rectangles =
null;

    public string LastName { get { return lastName; } }
    public string FirstName { get { return firstName; } }
    public string MiddleName { get { return middleName; } }
    public string PassportNo { get { return passportNo; } }
    public string Authority { get { return authority; } }
    public DateTime Birthday { get { return birthday; } }
    public DateTime DateOfIssue { get { return dateOfIssue; } }
    public DateTime DateOfExpire { get { return dateOfExpire; } }
}

    public bool Sex { get { return sex; } }
}

```

```

    public bool IsUnlimited { get { return isUnlimited; } }
    public bool IsClosed { get { return isClosed; } }
    public Image Picture { get { return picture; } }
    public List<Dictionary<string, Rectangle>> Rectangles { get
{ return rectangles; } }

    void fillList()
    {
        //Simple Passport
        Dictionary<string, Rectangle> pRect = new
Dictionary<string, Rectangle>();
        pRect.Add("lastname", new Rectangle(new Point(136, 70),
new Size(564, 100)));
        pRect.Add("firstname", new Rectangle(new Point(136,
224), new Size(564, 76)));
        pRect.Add("middlename", new Rectangle(new Point(136,
300), new Size(564, 74)));
        pRect.Add("birthday", new Rectangle(new Point(136, 444),
new Size(564, 78)));
        pRect.Add("dateOfIssue", new Rectangle(new Point(136,
522), new Size(564, 72)));
        pRect.Add("dateOfExpire", new Rectangle(new Point(790,
530), new Size(350, 74)));
        pRect.Add("sex", new Rectangle(new Point(1020, 160), new
Size(200, 80)));
        pRect.Add("passportNo", new Rectangle(new Point(1050,
1100), new Size(400, 100)));
        pRect.Add("authority", new Rectangle(new Point(136,
760), new Size(1284, 74)));
        pRect.Add("picture", new Rectangle(new Point(85, 1235),
new Size(400, 535)));

        pRect.Add("lastnameEng", new Rectangle(new Point(500,
1190), new Size(700, 70)));
        pRect.Add("firstnameEng", new Rectangle(new Point(500,
1260), new Size(700, 75)));
        pRect.Add("birthdayEng", new Rectangle(new Point(500,
1415), new Size(420, 75)));
        pRect.Add("dateOfIssueEng", new Rectangle(new Point(500,
1565), new Size(420, 75)));
        pRect.Add("dateOfExpireEng", new Rectangle(new
Point(980, 1565), new Size(480, 100)));
        pRect.Add("sexEng", new Rectangle(new Point(980, 1415),
new Size(170, 85)));

        //BioPassport
        Dictionary<string, Rectangle> bRect = new
Dictionary<string, Rectangle>();
        bRect.Add("lastname", new Rectangle(new Point(70, 320),
new Size(730, 85)));

```

```

        bRect.Add("firstname", new Rectangle(new Point(70, 405),
new Size(730, 90)));
        bRect.Add("middlename", new Rectangle(new Point(70,
495), new Size(730, 90)));
        bRect.Add("birthday", new Rectangle(new Point(70, 585),
new Size(310, 80)));
        bRect.Add("dateOfIssue", new Rectangle(new Point(500,
1620), new Size(350, 65)));
        bRect.Add("dateOfExpire", new Rectangle(new Point(500,
1685), new Size(350, 75)));
        bRect.Add("sex", new Rectangle(new Point(1145, 420), new
Size(175, 100)));
        bRect.Add("passportNo", new Rectangle(new Point(1060,
1090), new Size(390, 90)));
        bRect.Add("authority", new Rectangle(new Point(70, 740),
new Size(730, 140)));
        bRect.Add("picture", new Rectangle(new Point(85, 1195),
new Size(400, 535)));

        bRect.Add("lastnameEng", new Rectangle(new Point(490,
1175), new Size(570, 95)));
        bRect.Add("firstnameEng", new Rectangle(new Point(490,
1280), new Size(570, 100)));
        bRect.Add("birthdayEng", new Rectangle(new Point(490,
1450), new Size(380, 65)));
        bRect.Add("dateOfIssueEng", new Rectangle(new Point(0,
0), new Size(0, 0)));
        bRect.Add("dateOfExpireEng", new Rectangle(new Point(0,
0), new Size(0, 0)));
        bRect.Add("sexEng", new Rectangle(new Point(490, 1515),
new Size(160, 75)));

        rectangles = new List<Dictionary<string, Rectangle>>();
        rectangles.Add(pRect);
        rectangles.Add(bRect);
    }

    public PleaseWaitForm(string imageLocation, bool
isBioPassport)
    {
        InitializeComponent();
        this.progressBar1.AutoHeight = true;
        this.imageLocation = imageLocation;
        this.isBioPassport = isBioPassport;
        fillList();
        Size size;
        if (isBioPassport)
            size = new Size(1488, 2043);
        else
            size = new Size(1520, 2056);
        if (!EditImage.ResizeImage(imageLocation, size))

```

```

        DialogResult =
System.Windows.Forms.DialogResult.Cancel;
    }

    #region Overrides

    public override void SetCaption(string caption)
    {
        base.SetCaption(caption);
        this.progressPanel1.Caption = caption;
    }
    public override void SetDescription(string description)
    {
        base.SetDescription(description);
        this.progressPanel1.Description = description;
    }
    public override void ProcessCommand(Enum cmd, object arg)
    {
        base.ProcessCommand(cmd, arg);
    }

    #endregion

    public enum WaitFormCommand
    {
    }

    private void PleaseWaitForm_Shown(object sender, EventArgs
e)
    {
        try
        {
            string temp =
TextFromImage.GetTextFromImage(Image.FromFile(imageLocation),
MODI.MiLANGUAGES.miLANG_RUSSIAN).ToLower();
            if (temp.Contains("фам")
                || temp.Contains("исм")
                || temp.Contains("от"))
                this.isBioPassport = false;
            else
                this.isBioPassport = true;
            ////Cropped Image Recognizing with MODI
            int isBP = 0;
            MODI.MiLANGUAGES language;
            string tSex = "АЁЛ";
            if (isBioPassport)
            {
                isBP = 1;
                language = MODI.MiLANGUAGES.miLANG_ENGLISH;
                tSex = "AYOL";
            }
        }
    }

```

```

else
{
    isBP = 0;
    language = MODI.MiLANGUAGES.miLANG_RUSSIAN;
    tSex = "АЁЛ";
}
string lNameR =
TextFromImage.GetTextFromImage(EditImage.CropImage(Image.FromFile(im
ageLocation), rectangles[isBP]["lastname"]),
language).Replace("ФАМИЛИЯ", "");
string lNameE =
TextFromImage.GetTextFromImage(EditImage.CropImage(Image.FromFile(im
ageLocation), rectangles[isBP]["lastnameEng"]),
MODI.MiLANGUAGES.miLANG_ENGLISH).Replace("FAMILIYA", "");
string fNameR =
TextFromImage.GetTextFromImage(EditImage.CropImage(Image.FromFile(im
ageLocation), rectangles[isBP]["firstname"]),
language).Replace("ИСМ", "");
string fNameE =
TextFromImage.GetTextFromImage(EditImage.CropImage(Image.FromFile(im
ageLocation), rectangles[isBP]["firstnameEng"]),
MODI.MiLANGUAGES.miLANG_ENGLISH).Replace("ISM", "");

    lastName = isBioPassport ?
EditString.GetCorrectString(LatinToCyrill.Convert(lNameR), lNameE) :
EditString.GetCorrectString(lNameR, lNameE);

//EditString.GetCorrectString(LatinToCyrill.Convert(lNameR),
lNameE);
    firstName = EditString.GetCorrectString(fNameR,
fNameE);
    middleName =
TextFromImage.GetTextFromImage(EditImage.CropImage(Image.FromFile(im
ageLocation), rectangles[isBP]["middlename"]), language).Replace("
", "");
    if (isBioPassport)
        middleName =
EditString.GetOnlyLetters(LatinToCyrill.Convert(middleName),
EditString.Alphabet.Cyrillic).Replace("ОТА", "").Replace("ОТА",
 "").Replace(" ", "");
    else
        middleName =
EditString.GetOnlyLetters(middleName,
EditString.Alphabet.Cyrillic).Replace("ОТАСИННИГ",
 "").Replace("ИСМИ", "").Replace("ОТЧЕСТВО", "").Replace("ОТА",
 "").Replace("ОТА", "").Replace(" ", "");
    string bDayR =
TextFromImage.GetTextFromImage(EditImage.CropImage(Image.FromFile(im
ageLocation), rectangles[isBP]["birthday"]), language);
    string bDayE =
TextFromImage.GetTextFromImage(EditImage.CropImage(Image.FromFile(im

```

```

ageLocation), rectangles[isBP]["birthdayEng"]),
MODI.MiLANGUAGES.miLANG_ENGLISH);
    if (isBioPassport)
    {
        bDayR = EditString.GetCorrectDateTime(bDayR);
        bDayE = EditString.GetCorrectDateTime(bDayE);
    }
    if (bDayR.Length == 10)
    {
        if (!DateTime.TryParse(bDayR, out birthday))
            DateTime.TryParse(bDayE, out birthday);
    }
    else
        DateTime.TryParse(bDayE, out birthday);
    string iDateR =
TextFromImage.GetTextFromImage(EditImage.CropImage(Image.FromFile(im
ageLocation), rectangles[isBP]["dateOfIssue"]), language);
    string iDateE =
TextFromImage.GetTextFromImage(EditImage.CropImage(Image.FromFile(im
ageLocation), rectangles[isBP]["dateOfIssueEng"]),
MODI.MiLANGUAGES.miLANG_ENGLISH);
    if (isBioPassport)
    {
        iDateR = EditString.GetCorrectDateTime(iDateR);
        iDateE = EditString.GetCorrectDateTime(iDateE);
    }
    if (!DateTime.TryParse(iDateR, out dateOfIssue))
        DateTime.TryParse(iDateE, out dateOfIssue);
    string eDateR =
TextFromImage.GetTextFromImage(EditImage.CropImage(Image.FromFile(im
ageLocation), rectangles[isBP]["dateOfExpire"]), language);
    string eDateE =
TextFromImage.GetTextFromImage(EditImage.CropImage(Image.FromFile(im
ageLocation), rectangles[isBP]["dateOfExpireEng"]),
MODI.MiLANGUAGES.miLANG_ENGLISH);
    if (isBioPassport)
    {
        eDateR = EditString.GetCorrectDateTime(eDateR);
        eDateE = EditString.GetCorrectDateTime(eDateE);
    }
    if (DateTime.TryParse(eDateR, out dateOfExpire))
        isUnlimited = false;
    else if (DateTime.TryParse(eDateE, out
dateOfExpire))
        isUnlimited = false;
    else
        isUnlimited = true;
    string seX =
EditString.GetLastWord(TextFromImage.GetTextFromImage(EditImage.Crop
Image(Image.FromFile(imageLocation), rectangles[isBP]["sex"]),
language));

```

```

        sex = seX == tSex ? false : true;
        string pNo =
TextFromImage.GetTextFromImage(EditImage.CropImage(Image.FromFile(imageLocation), rectangles[isBP]["passportNo"]),
MODI.MiLANGUAGES.miLANG_ENGLISH).Replace(" ", "");
        passportNo = pNo.Substring(pNo.Length - 9).Insert(2,
" ");
        EditString.Alphabet alphabet = isBioPassport == true
? EditString.Alphabet.Latin : EditString.Alphabet.Cyrillic;
        authority =
EditString.GetOnlyLetters(TextFromImage.GetTextFromImage(EditImage.C
ropImage(Image.FromFile(imageLocation),
rectangles[isBP]["authority"]), language), alphabet);
        if (isBioPassport)
            authority = LatinToCyrill.Convert(authority);
        picture =
EditImage.CropImage(Image.FromFile(imageLocation),
rectangles[isBP]["picture"]);
    }
    catch (Exception ex)
    {
        //MessageBox.Show(ex.ToString());
    }
    isClosed = true;
    DialogResult = System.Windows.Forms.DialogResult.OK;
}
}
}

```

д) Распознавание лиц с видеопотока

```

public class EigenObjectRecognizer
{
    private Image<Gray, Single>[] _eigenImages;
    private Image<Gray, Single> _avgImage;
    private Matrix<float>[] _eigenValues;
    private string[] _labels;
    private double _eigenDistanceThreshold;

    public Image<Gray, Single>[] EigenImages
    {
        get { return _eigenImages; }
        set { _eigenImages = value; }
    }

    /// <summary>
    /// Get or set the labels for the corresponding training image
    /// </summary>
    public String[] Labels
    {
        get { return _labels; }
    }
}

```

```

        set { _labels = value; }
    }

    public double EigenDistanceThreshold
    {
        get { return _eigenDistanceThreshold; }
        set { _eigenDistanceThreshold = value; }
    }

    public Image<Gray, Single> AverageImage
    {
        get { return _avgImage; }
        set { _avgImage = value; }
    }

    public Matrix<float>[] EigenValues
    {
        get { return _eigenValues; }
        set { _eigenValues = value; }
    }

    private EigenObjectRecognizer()
    {
    }

    public EigenObjectRecognizer(Image<Gray, Byte>[] images, ref
MCvTermCriteria termCrit)
        : this(images, GenerateLabels(images.Length), ref termCrit)
    {
    }

    private static String[] GenerateLabels(int size)
    {
        String[] labels = new string[size];
        for (int i = 0; i < size; i++)
            labels[i] = i.ToString();
        return labels;
    }

    public EigenObjectRecognizer(Image<Gray, Byte>[] images,
String[] labels, ref MCvTermCriteria termCrit)
        : this(images, labels, 0, ref termCrit)
    {
    }

    public EigenObjectRecognizer(Image<Gray, Byte>[] images,
String[] labels, double eigenDistanceThreshold, ref MCvTermCriteria
termCrit)
    {
        Debug.Assert(images.Length == labels.Length, "The number of
images should equals the number of labels");
    }

```

```

        Debug.Assert(eigenDistanceThreshold >= 0.0, "Eigen-distance
threshold should always >= 0.0");

        CalcEigenObjects(images, ref termCrit, out _eigenImages,
out _avgImage);

        _eigenValues = Array.ConvertAll<Image<Gray, Byte>,
Matrix<float>>(images,
                delegate(Image<Gray, Byte> img)
                {
                    return new Matrix<float>(EigenDecomposite(img,
_eigenImages, _avgImage));
                });

        _labels = labels;

        _eigenDistanceThreshold = eigenDistanceThreshold;
    }

    #region static methods
    public static void CalcEigenObjects(Image<Gray, Byte>[]
trainingImages, ref MCvTermCriteria termCrit, out Image<Gray,
Single>[] eigenImages, out Image<Gray, Single> avg)
    {
        int width = trainingImages[0].Width;
        int height = trainingImages[0].Height;

        IntPtr[] inObjs = Array.ConvertAll<Image<Gray, Byte>,
IntPtr>(trainingImages, delegate(Image<Gray, Byte> img) { return
img.Ptr; });

        if (termCrit.max_iter <= 0 || termCrit.max_iter >
trainingImages.Length)
            termCrit.max_iter = trainingImages.Length;

        int maxEigenObjs = termCrit.max_iter;

        #region initialize eigen images
        eigenImages = new Image<Gray, float>[maxEigenObjs];
        for (int i = 0; i < eigenImages.Length; i++)
            eigenImages[i] = new Image<Gray, float>(width, height);
        IntPtr[] eigObjs = Array.ConvertAll<Image<Gray, Single>,
IntPtr>(eigenImages, delegate(Image<Gray, Single> img) { return
img.Ptr; });
        #endregion

        avg = new Image<Gray, Single>(width, height);

        CvInvoke.cvCalcEigenObjects(
            inObjs,
            ref termCrit,

```

```

        eigObjs,
        null,
        avg.Ptr);
    }

    public static float[] EigenDecomposite(Image<Gray, Byte> src,
Image<Gray, Single>[] eigenImages, Image<Gray, Single> avg)
    {
        return CvInvoke.cvEigenDecomposite(
            src.Ptr,
            Array.ConvertAll<Image<Gray, Single>,
IntPtr>(eigenImages, delegate(Image<Gray, Single> img) { return
img.Ptr; })),
            avg.Ptr);
    }
    #endregion

    public Image<Gray, Byte> EigenProjection(float[] eigenValue)
    {
        Image<Gray, Byte> res = new Image<Gray,
byte>(_avgImage.Width, _avgImage.Height);
        CvInvoke.cvEigenProjection(
            Array.ConvertAll<Image<Gray, Single>,
IntPtr>(_eigenImages, delegate(Image<Gray, Single> img) { return
img.Ptr; })),
            eigenValue,
            _avgImage.Ptr,
            res.Ptr);
        return res;
    }

    public float[] GetEigenDistances(Image<Gray, Byte> image)
    {
        using (Matrix<float> eigenValue = new
Matrix<float>(EigenDecomposite(image, _eigenImages, _avgImage)))
            return Array.ConvertAll<Matrix<float>,
float>(_eigenValues,
                delegate(Matrix<float> eigenValueI)
                {
                    return (float)CvInvoke.cvNorm(eigenValue.Ptr,
eigenValueI.Ptr, Emgu.CV.CvEnum.NORM_TYPE.CV_L2, IntPtr.Zero);
                });
    }

    public void FindMostSimilarObject(Image<Gray, Byte> image, out
int index, out float eigenDistance, out String label)
    {
        float[] dist = GetEigenDistances(image);

        index = 0;
        eigenDistance = dist[0];
    }

```

```

    for (int i = 1; i < dist.Length; i++)
    {
        if (dist[i] < eigenDistance)
        {
            index = i;
            eigenDistance = dist[i];
        }
    }
    label = Labels[index];
}

public String Recognize(Image<Gray, Byte> image)
{
    int index;
    float eigenDistance;
    String label;
    FindMostSimilarObject(image, out index, out eigenDistance,
out label);

    return (_eigenDistanceThreshold <= 0 || eigenDistance <
_eigenDistanceThreshold ) ? _labels[index] : String.Empty;
}
}

```