

Министерство образования Республики Беларусь  
Учреждение образования  
Белорусский государственный университет  
информатики и радиоэлектроники  
Кафедра инженерной психологии и эргономики

На правах рукописи

УДК

Галай  
Дмитрий Александрович

ПОВЫШЕНИЕ ЭРГОНОМИЧЕСКИХ ПОКАЗАТЕЛЕЙ ПРОГРАММНОГО  
СРЕДСТВА РАСПОЗНАВАНИЯ ТЕКСТОВОЙ ИНФОРМАЦИИ НА  
ИЗОБРАЖЕНИИ

Автореферат на соискание академической степени  
магистра технических наук

1 - 23 80 08 Психология труда, инженерная психология, эргономика

Магистрант Д.А. Галай

Научный руководитель  
И.Л. Качалов, кандидат  
исторических наук, доцент

Минск 2019

## ВВЕДЕНИЕ

Несмотря на то, что в настоящее время большинство документов составляется при помощи компьютеров, задача создания полностью электронного документооборота все еще далека до своей полной реализации. Как правило, существующие компьютерные системы охватывают деятельность лишь отдельных организаций, а непосредственный обмен данными между организациями осуществляется при помощи традиционных бумажных документов.

Задача перевода информации с бумажных носителей на электронные на сегодняшний день актуальна не только в рамках потребностей, возникающих в системах документооборота. Существующие в современном мире информационные технологии позволяют существенно упростить доступ к информационным ресурсам, накопленным человечеством в течении нескольких десятков лет, при том условии, что они будут переведены в электронный вид.

Так наиболее простым и быстрым способом перевода документов в электронный вид является их сканирование при помощи сканера, результатом работы которого является цифровое изображение документа – графический файл. Наиболее удобным, по сравнению с графическим, является текстовое представление информации. Такой способ представления информации позволяет существенно сократить затраты на ее хранение и передачу, а также позволяет реализовать все возможные сценарии использования и анализа электронных документов: редактирование, форматирование, перевод, поиск и т.д. Поэтому наибольший интерес с практической точки зрения представляет именно перевод бумажных носителей в текстовый электронный документ.

Процесс автоматического перевода текстовой информации с бумажных или графических носителей в текстовый электронный документ связан с задачей распознавания образов, а именно строк текста и символов. Большое количество шрифтов, способов форматирования и написания текста усложняют задачу распознавания. В связи с этим для решения подобных задач все чаще прибегают к использованию искусственных нейронных сетей, представляющих собой математические модели, построенные по принципу организации и функционирования биологических нейронных сетей.

Нейронные сети способны решать задачи, в которых неизвестны закономерности развития ситуации и зависимости между входными и выходными данными, что делает их использование более приемлемым по

сравнению с традиционными математическими методами или экспертными системами. В задачах распознавания выбор в сторону использования нейронных сетей часто делается и потому, что они наименее чувствительны к неинформативным или шумовым входным сигналам.

На сегодняшний день существует немало способов произвести перевод машинного текста с бумажных носителей в электронно-цифровой вид. Большинство таких способов заключается в использовании специальных программных средств. Однако, как правило, большая часть таких программ имеют ряд общих недостатков, среди которых особенно стоит выделить следующие: отсутствие интуитивности пользовательского интерфейса, избыточность элементов управления программным средством, отсутствие какой-либо документации и т.д.

## Понятие распознавания текста

Тема распознавания текста является одной из основных тем распознавания образов. 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 других объектов, которые характеризуются конечным набором некоторых свойств и признаков. Такие задачи решаются довольно часто, например, при переходе или проезде улицы по сигналам светофора. Распознавание цвета загоревшейся лампы светофора и знание правил дорожного движения позволяет принять правильное решение о том, можно или нельзя переходить улицу.

Необходимость в таком распознавании возникает в самых разных областях: от военного дела и систем безопасности до оцифровки аналоговых сигналов.

Проблема распознавания образа приобрела выдающееся значение в условиях информационных перегрузок, когда человек не справляется с линейно-последовательным пониманием поступающих к нему сообщений, в результате чего его мозг переключается на режим одновременности восприятия и мышления, которому такое распознавание свойственно.

Неслучайно, таким образом, проблема распознавания образа оказалась в поле междисциплинарных исследований – в том числе в связи с работой по созданию искусственного интеллекта, а создание технических систем распознавания образа привлекает к себе всё большее внимание.

В теме распознавания образов существует два основных направления:

- изучение способностей к распознаванию, которыми обладают живые существа, их объяснение и моделирование;
- развитие теории и методов построения устройств и программных средств, предназначенных для решения отдельных задач по распознаванию образов в прикладных целях.

Распознавание текста – одна из основных областей применения систем распознавания образов. Под распознаванием текста подразумевается процесс перевода изображений, содержащих рукописный, машинный или печатный текст в текстовые данные, используемые для представления символов в компьютере (например, в текстовом редакторе).

Распознавание текста на изображении – сложный процесс, который, как правило, состоит из следующих более простых и независимых друг от друга этапов (стадий):

- шумоподавление;
- бинаризация;
- выделение строк;
- выделение символов;
- выбор шрифта;
- распознавание.

### **Алгоритм распознавания текста**

Первый шаг алгоритма включает в себя удаление возможных шумов на изображении, полученных в результате не идеальности устройств, породивших его. Процесс шумоподавления осуществляется при помощи фильтров для изображений. Наиболее благоприятный результат с минимальными машинными затратами в этом можно достичь, применив median-фильтр.

На стадии бинаризации изображение приводится к такому виду, при котором значение каждого его элемента становится равным 0 или 1, то есть бинарным. В общем случае преобразование осуществляется над общим значением яркости изображения, то есть результирующее изображение будет полностью черно-белым. Бинаризация применяется к изображению разделения значимых пикселей от незначимых. Под значимыми пикселями подразумеваются те пиксели, которые на изображении соответствуют тексту.

Любой символ на изображении обладает своими характеристиками, которые в своей совокупности являются уникальными для каждого символа. К таким характеристикам символа относятся его периметр, площадь, плотность, удлиненность, ориентация, количество углов и другие. Одни и те же символы обладают схожими значениями таких характеристик, а значит их знание для символов, входящих в алфавит некоторого языка, может помочь определить значение неизвестного символа по его найденным характеристикам, путем сравнения их с уже известными.

Процесс вычисления геометрических параметров каждого символа упрощается благодаря полученным на предыдущем шаге связанным областям. Так, например, для вычисления площади символа достаточно определить

количество пикселей, входящих в соответствующую ему на изображении связанную область. Стоит отметить, что при работе с разными изображениями стоит учитывать размер каждого из них (ширина и высота) и, если это необходимо, предварительно производить соответствующие операции масштабирования.

Вычисленные параметры каждого символа позволяют определить значение самого символа при помощи нейронной сети. Для этого, перед самим процессом определения символа, нужно иметь нейронную сеть, которая заранее натренирована на распознавание символов алфавита конкретного языка.

Подразумевается, что перед запуском программного средства нейронная сеть уже натренирована на распознавание символов конкретного алфавита и объект, представляющий ее, инициализируется каждый раз перед запуском программы. Чтобы избавиться от необходимости производить обучение нейронной сети каждый раз перед запуском программы, было решено хранить сериализованный объект нейронной сети в файле (таким образом, файл содержит структуру данных, описывающую нейронную сеть, как последовательность бит) и загружать его каждый раз при запуске программы. Такой подход позволяет сократить время на инициализацию объектов при запуске программы, а также добавить возможность обновлять существующие и добавлять новые файлы, добавляющие поддержку новых языков.

В качестве используемых в программном средстве нейронных сетей были выбраны нейронные сети прямого распространения, состоящие их трех слоев (слой входных нейронов, слой скрытых нейронов и слой выходных нейронов).

Сети прямого распространения – это такие сети, все связи которых направлены строго от входных нейронов к выходным. Данный тип сетей обладает высокой точностью, быстрой скоростью работы и несложной реализацией.

Процесс обучения нейронной сети прямого распространения заключается в передаче на ее вход некоторых данных (обучающей выборки) и их единоразовый просмотр. В ходе просмотра нейронную сеть обучающей выборки происходит перерасчет ее внутренних коэффициентов (весов), что в конечном счете влияет на результат ее работы.

Данные, подаваемые на вход нейронной сети, представляют собой вычисленные параметры того или иного символа некоторого алфавита. В ходе экспериментов было установлено, что сеть способна удовлетворительно определять значение символа по следующим его геометрическим характеристикам:

- общая площадь символа;
- площадь символа, входящего в определенную зону (всего имеется четыре зоны, полученные путем деления области символа пополам вдоль горизонтальной и вертикальной осей);
- периметр;
- плотность;
- удлиненность;
- ориентация;
- количество углов.

Таким образом, общее количество входных параметров нейронной сети равно общему количеству параметров символа, то есть десяти.

Количество выходов нейронной сети равно количеству символов в языке конкретного алфавита. Выход, на котором в результате работы сети окажется максимальное значение, соответствует значению символа.

## **Описание модулей программного средства**

Архитектура программного обеспечения – это структура программы, которая включает программные компоненты, видимые снаружи свойства этих компонентов, а также отношения между ними.

Особенностью рассматриваемого программного средства является то, что по своей сути оно состоит из шести слабосвязанных частей или модулей:

- 1 Модуль, отвечающий за возможность работы с фильтрами для обработки изображений.
2. Модуль, отвечающий за возможность редактирования изображений.
3. Модуль, отвечающий за возможность распознавания машинного текста на изображениях.
4. Модуль, содержащий функционал для работы с нейронной сетью.
5. Модуль, предоставляющий пользователям графический интерфейс и координирующий работу остальных модулей.
6. Модуль, содержащий вспомогательные типы данных, которые используются в других модулях.

Для упрощения дальнейшей разработки и возможности расширения программного средства в будущем было принято решение строить его при

помощи нескольких слабосвязанных модулей на основе принципов проектирования SOLID.

Разделение программного средства на такое количество модулей связано с несколькими причинами:

- ограничение области ответственности каждого модуля одной узкоспециализированной задачей;

- удобство тестирования функциональности каждого модуля отдельно от остальных;

- возможность замены в будущем одного модуля другим без внесения или внесения незначительных изменений в другие модули, используемые замененный (подразумевается, что новый модуль сохранит интерфейс старого модуля);

- возможность использования определенных отдельных модулей для решения аналогичных задач в других программных средствах;

- возможность нескольким разработчикам вести разработку программного средства параллельно, не влияя на работу друг друга.

Взаимодействие между всеми модулями программного средства осуществляется при помощи заранее оговоренного интерфейса, который в процессе разработки не изменяется или изменяется незначительно только в редких случаях.

## **Модуль для работы с фильтрами изображений**

В результате анализа фильтров изображений, необходимых для работы программного средства, было определено, что все используемые в дальнейшем фильтры в зависимости от алгоритма их работы можно условно разделить на две группы:

- фильтры, применяемые к отдельному пикселю на изображении (grayscale-фильтр, negative-фильтр и другие);

- фильтры, использующие матрицу свертки (min-фильтр, max-фильтр, min-max-фильтр, median-фильтр и другие).

К первой группе фильтров относятся такие фильтры, результат работы которых для заданного пикселя будет зависеть только от параметров самого пикселя. К таким параметрам относятся составляющие красного, зеленого и синего цветов, а также значение яркости пикселя.



Ко второй же группе фильтров относятся такие фильтры, результат работы которых для заданного пикселя будет зависеть как от параметров самого пикселя, так и параметров пикселей, расположенных на сравнительно малом удалении от него, образующих при этом квадрат (матрицу), центром которого является заданный пиксель.

Фильтры, входящие в конкретную группу, имеют лишь небольшие отличия друг от друга. Так, например, для фильтров, входящих в первую группу, отличными будут лишь коэффициенты, применяемые к каждой цветовой составляющей исходного пикселя, а для фильтров, входящих во вторую группу, отличными будут лишь операции обработки матрицы свертки исходного пикселя.

В связи с этим было принято решение о выделении базового интерфейса «IFilter» для всех фильтров и производных от него интерфейсов «IPixelFilter» и «IConvolutionFilter» для фильтров из первой и второй групп соответственно. Общий интерфейс «IFilter» необходим, поскольку использование рассматриваемого модуля верхними модулями, не подразумевает их разделение на две группы.

На основе созданных интерфейсов были спроектированы абстрактные классы «BaseFilter», «BasePixelFilter» и «BaseConvolutionFilter», последние два из которых реализует логику обхода пикселей изображения в зависимости от принадлежности к первой или второй группе. При этом ответственность за вычисление результирующего значения для очередного пикселя лежит на классах, производных от «BasePixelFilter» или «BaseConvolutionFilter». Класс «BaseFilter», в свою очередь, содержит общую часть всех фильтров. Так, например, в нем содержится реализация механизма обработки ошибок, являющаяся одинаковой для всех фильтров.

## **Модуль для редактирования изображений**

В результате анализа операций редактирования изображений, которые должны иметь место в программном средстве, было решено спроектировать на каждую операцию по отдельному интерфейсу.

В результате чего были получены интерфейсы «IRotator» для операции поворота изображения, «IReflector» для операции отражения изображения и «IResizer» для операции обрезки изображения. Каждому интерфейсу

соответствует свой класс, реализующий конкретный механизм редактирования изображений.

## **Модуль распознавания текста на изображении**

В результате анализа требований к процессу распознавания текста на изображении, было решено спроектировать интерфейс «ITextRecognizer», предоставляющий методы для распознавания текста на изображении. На основе этого интерфейса был реализован класс «TextRecognizer», который выполнения внутренней логики использует классы из модуля, содержащего функционал для работы с нейронной сетью.

В результате анализа литературных источников было определено, что нейронная сеть должна выполнять два основных процесса: обучение и определение принадлежности заданного образа к некоторому классу. В связи с этим было решено спроектировать интерфейс «INeuralNetwork» и его реализацию в виде класса «NeuralNetwork».

## **Модуль предоставляющий графический интерфейс**

Модуль, предоставляющий пользователям графический интерфейс и координирующий работу остальных модулей, по своей сути расположен на уровне представления разрабатываемого программного средства.

Уровень представления разрабатываемого программного средства представлен программным модулем, основой которого является платформа Windows Presentation Foundation (WPF). В основе WPF лежит векторная система визуализации, не зависящая от разрешения устройства вывода и созданная с учётом возможностей современного графического оборудования.

WPF предоставляет средства для создания визуального интерфейса, включая язык XAML (Extensible Application Markup Language), элементы управления, привязку данных, макеты, двухмерную и трёхмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление.

XAML представляет собой язык декларативного описания интерфейса, основанный на XML [22]. Также реализована модель разделения кода и дизайна,

позволяющая кооперироваться программисту и дизайнеру. Кроме того, есть встроенная поддержка стилей элементов, а сами элементы легко разделить на элементы управления второго уровня, которые, в свою очередь, разделяются до уровня векторных фигур и свойств/действий. Это позволяет легко задать стиль для любого элемента, например, Button (кнопка).

Графической технологией, лежащей в основе WPF, является DirectX, в отличие от Windows Forms, где используется GDI/GDI+. Производительность WPF выше, чем у GDI+ за счёт использования аппаратного ускорения графики через DirectX.

При разработке архитектуры программного средства было обнаружено, что многие модули содержат одинаковый с точки зрения проектирования и программной реализации функционал (например, механизм обработки ошибочных или исключительных ситуаций).

Чтобы избавиться от возможного повторения одного и того же программного кода в разных частях программного средства, было решено спроектировать еще один вспомогательный модуль, содержащий используемые остальными модулями типы.

## ЗАКЛЮЧЕНИЕ

В рамках данной работы были повышены эргономические показатели программного средства распознавания текстовой информации на изображении при помощи нейронной сети.

Данный проект является результатом процесса, включающего всесторонний анализ поставленной задачи, исследование существующих аналогов, работу со справочной литературой и интернет-источниками, дизайн графического интерфейса пользователя, разработку алгоритмов, тестирование всех функций программы.

Программа предоставляет пользователю все требуемые согласно поставленной задаче возможности.

Приложение было протестировано на различных (корректных и некорректных) наборах данных, и, как было показано в пункте «Тестирование, экспериментальное исследование и анализ полученных результатов», полученные результаты в общем случае соответствуют ожидаемым.

Опрос пользователей программы показал, что в результате работы эргономические показатели программного средства были улучшены.

На основании вышеперечисленного можно утверждать, что созданное приложение соответствует всем поставленным условиям и в процессе разработки были учтены все требования, сформулированные при постановке задачи.