[1 СУЩЕСТВУЮЩИЕ МЕТОДЫ РАСПОЗНАВАНИЯ ГРАФИЧЕСКИХ ОБРАЗОВ 2](#_Toc489612140)

[2 НЕЙРО-НЕЧЕТКИЙ КЛАССИФИКАТОР NEFCLASS 4](#_Toc489612141)

[2.1 Архитектура 4](#_Toc489612142)

[2.2 Генерация базы продукционных правил 5](#_Toc489612143)

[2.3 Настройка параметров функции принадлежности 7](#_Toc489612144)

[3 ОБРОБКА ЗОБРАЖЕННЯ 9](#_Toc489612145)

[3.1 Конвертация цветоного изображения в монохромное 9](#_Toc489612146)

[3.2 Выделение изображения цифры 10](#_Toc489612147)

[3.3 Медианная фильтрация 11](#_Toc489612148)

[3.4 Разделение изображения цифры на области 12](#_Toc489612149)

[4 ОПИСАНИЕ ПРОГРАММЫ 15](#_Toc489612150)

[4.1 Алгоритм работы программы 15](#_Toc489612151)

[4.2 Работа с программой 16](#_Toc489612152)

[5 РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ 18](#_Toc489612153)

[5.1 База правил 18](#_Toc489612154)

[5.2 Распознавание 18](#_Toc489612155)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 19](#_Toc489612156)

[БЛОК-СХЕМЫ 20](#_Toc489612157)

1 СУЩЕСТВУЮЩИЕ МЕТОДЫ РАСПОЗНАВАНИЯ ГРАФИЧЕСКИХ ОБРАЗОВ

Решение задачи распознавания последовательности символов (например, цифр) можно представить в виде решения задачи классификации.

Существуют метрические алгоритмы классификации, в том числе, и для распознавания текста. Группа методов на основе метрик, например, метрики Хэмминга, показывают, насколько объекты не похожи между собой. Данная метрика часто используется при кодировании информации и передаче данных. Вычисление метрики происходит путем сравнения частей последовательности с соответствующими частями из другой последовательности. В результате по количеству позиций, в которых объекты не соответствуют друг другу, вычисляется степень непохожести. Следовательно, чтобы определить какая цифра изображена, нужно найти ее метрику со всеми готовыми шаблонами. И тот шаблон, чья степень непохожести окажется минимальной, будет решением задачи. К недостаткам метрических алгоритмов стоит отнести тот факт, что подсчет одной лишь метрики не дает положительного результата, так как многие символы и, в том числе, цифры, похожи между собой: например «1» и «7», что приводит к ошибочному распознаванию.

Повысить качество классификации данных позволяют обучающиеся методы, например, метод опорных векторов и метод релевантных векторов. Метод опорных векторов заключается в построении классифицирующей функции, при помощи которой объекты классифицируются только на два класса, поэтому обучение для данного метода происходит быстрее, чем для других обучающихся методов (например, методов, в основе которых лежат нейронные сети), но при применении метода опорных векторов для решения многоклассовой задачи, качество и скорость работы данного метода снижаются.

Поэтому с учётом потенциальной возможности создания системы распознавания не только цифровых, но и символьных последовательностей другого типа с различным шрифтом, фоном и качеством написания, авторами был выбран подход к классификации данных на основе нейронных сетей с использованием теории нечётких множеств.

Система нечеткой классификации NEFCLASS (NEuro-Fuzzy CLASSifier), основана на архитектуре нечеткого персептрона и алгоритмах нечеткого логического вывода. В модифицированной системе NEFCLASSM используется гауссова функция принадлежности, алгоритм обучения нейро-нечёткой сети осуществляется на основе методов многомерной оптимизации, например, на основе метода сопряженных градиентов.

2 НЕЙРО-НЕЧЕТКИЙ КЛАССИФИКАТОР NEFCLASS

2.1 Архитектура

Модель NEFCLASS (NEuro Fuzzy CLASSifier) является нейро-нечётким классификатором, принцип работы которого основан на получении нечетких правил из множества данных, которые можно разделить на некоторое количество непересекающихся классов. Задача NEFCLASS состоит в том, чтобы определить принадлежность к классу образца с набором входных параметров. Нечеткость объясняется несовершенством или неполнотой измерений тех свойств объектов, по которым осуществляется процесс классификации.

Система NEFCLASS имеет 3-слойную последовательную архитектуру (рис. 3.1).

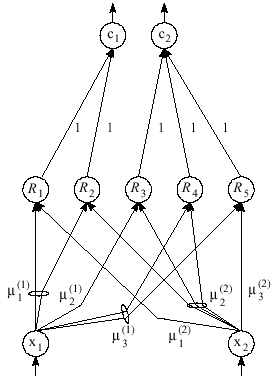


Рисунок 3.1 – Пример система NEFCLASS в виде 3-слойной нейронной сети  
с двумя входными и двумя выходными нейронами.

Первый слой содержит входные нейроны. Второй, скрытый, слой содержит продукционные правила, каждое из которых представлено в следующем виде:

ЕСЛИ x1 является µ1 И x2 является µ2 И … И xn является µn , ТО образец с вектором параметров  принадлежит классу ci,

где µ1… µn - нечеткие множества,

x1… xn - входные параметры классификатора

Третий слой состоит из выходных нейронов, каждый из которых соответствует определённому классу. Активация для нейрона правила R в сети с n входными нейронами вычисляется следующим образом:

где - нечеткий вес соединения входного нейрона *хi* с нейроном правила R.

Активация для нейрона выходного слоя, связанного с m нейронами правил скрытого слоя вычисляется следующим образом:

Соотношения (3.1) и (3.2) определяются путем замены нечётких термов, качественно характеризующих значения входных параметров, на функции принадлежности значения каждого входного нейрона каждому нечёткому множеству, а также замены логических операций «И» и «ИЛИ» на функции t-нормы и s-нормы (t-конормы), например, на операции минимума и максимума.

2.2 Генерация базы продукционных правил

Система NEFCLASS может быть построена по частичным знаниям о классифицируемом объекте. Нужно определить количество начальных нечетких множеств и задать значение kmax – максимальное число нейронов правил, которые могут быть созданы в скрытом слое. Для обучения может быть использована гауссова функция принадлежности.

При генерации продукционных правил для системы NEFCLASS с n входными нейронами x1…xn, нейронами правил в количестве k≤kmax и m выходными нейронами c1…cm, используется обучающая выборка изображений, представленная множеством образцов с входными параметрами x1…xn и желаемым выходным параметром сi, характеризующим класс, к которому принадлежит данный образец обучающей выборки Обучающий алгоритм, цель которого заключается в формировании скрытого слоя нейронов правил системы NEFCLASS, состоит из следующих этапов:

1. Получение и обработка текущего изображения из обучающей выборки.
2. Нахождение для каждого входного нейрона xi такой функции принадлежности, которая соответствует соотношению:

где - значение функции принадлежности входного нейрона xi нечёткому множеству Aj,

p – количество нечётких множеств.

1. Если текущее количество нейронов правил меньше kmax и не существует нейрона правила R с вектором весовых коэффициентов, удовлетворяющим соотношению:

то создаем такой узел и соединяем его с выходным узлом сi, соответствующим желаемому выходному нейрону.

1. Если в обучающей выборке остались необработанные изображения, осуществляется переход к шагу 1, иначе переход к шагу 5.
2. Результирующая база продукционных правил определяется согласно процедуре “лучшего” обучение правил, в процессе которой при обработке образцов обучающей выборки происходит накапливание активации каждого нейрона правил для каждого класса образцов, которые были рассмотрены. Если нейрон правила R показывает большее накопление активации для класса сj, чем для класса cR, который был специфицирован для следствия правила, тогда следует изменить следствие правила R на сj, то есть соединить нейрон скрытого слоя R с нейроном выходного слоя сj. Обработка образцов из обучающей выборки производится для каждого нейрона скрытого слоя R путём вычисления накопления активации согласно соотношению:

где =1 в том случае, если *i*-й образец классифицирован верно, или =-1 в том случае, если *i*-й образец классифицирован не верно.

В результате обучения в нечёткой базе продукционных правил остается k нейронов с наивысшими значениями функции накопления, рассчитанной по уравнению (3.5).

2.3 Настройка параметров функции принадлежности

В модифицированной системе нечеткой классификации NEFCLASSM нечеткие множества имеют гауссову функцию принадлежности, описываемую согласно уравнению:

где a – параметр функции принадлежности, называемый коэффициентом максимума,

b – параметр функции принадлежности, называемый коэффициентом концентрации.

Задача настройки параметров функций принадлежности сводится к решению задачи минимизации функции среднеквадратичной ошибки на обучающей выборке:

где N – количество образцов обучающей выборки,

– вектор активации нейронов выходного слоя для обучающего образца p,

– целевой вектор активации нейронов выходного слоя для обучающего образца p,

– векторы параметров функций принадлежности.

Координаты целевого вектора для образца pi определяются согласно выражению:

где j – номер класса, которому принадлежит данный образец.

Аргументами численной оптимизации функции (3.7) является совокупный вектор параметров функций принадлежности всех нечетких множеств. Для решения задачи применяется метод сопряженных градиентов

3 ОБРОБКА ЗОБРАЖЕННЯ

3.1 Конвертация цветоного изображения в монохромное

При компьютерной обработке получаемая информация графического изображения сохраняется в цифровом виде по цветовой моделии RGB. Модель базируется на сочетании трех цветов – красный, зеленый и синий. В изображениях точкам отводится соответствующая пропорция смешивания этих цветов. И под каждый цвет выделяется фиксированная память восемь бит.

Любое цветовое пространство RGB должно быть связано с эталонным цветовым пространством CIE XYZ[6]. Для CIE XYZ известно соответствие между значением коэффициентов (x,y,z) пикселя и фактическим значением яркости эталонных источников света, смешением которых добиваются получения нужного воспринимаемого цвета. Для каждого RGB-изображения явно или неявно такое преобразование в XYZ должно быть определено, иначе оно может быть корректно показано только на том мониторе, на котором создавалось. Назовем такую информацию цветовым профилем изображения. Если такой профиль есть у монитора, то путем преобразования *RGBimage*→*XYZimage* и затем преобразования *XYZimage*→*RGBdisplay* монитора можно получить на другом мониторе цвета, соответствующие исходным (с учетом ограничений цветопередачи).

В пространстве XYZ координата Y по определению соответствует воспринимаемой яркости цвета. Чтобы получить из полноцветного изображения монохромное, необходимо преобразовать каждый пиксель в XYZ и взять компоненту Y в качестве результата. Известно, что преобразование между любыми аддитивными цветовыми системами линейное (в силу линейности восприятия цвета человеком), а, значит, может быть описано матрицей M3×3, такой, что .

Подавляющее большинство цифровых изображений на сегодняшний день соответствуют стандарту sRGB, так же необходимо выполнять гамма-преобразование[6]. Таким образом преобразование из цветного sRGB-изображения в монохромное выглядит так:

Коэффициенты (0.21, 0.72, 0.07) —  это округленная до второго знака строчка из матрицы преобразования *M sRGB* из системы sRGB в XYZ.

3.2 Выделение изображения цифры

Предполагается, что на изображении с распознаваемыми цифрами цвет этих цифр сильно отличается (выделяется) на цвете фона: например, в книге фон – белый, цифра – черная; на маркировке грузового вагона фон будет темным, цифра (маркировка) белая и т.д. Поэтому целесообразно работать с монохромным изображением (цвета изображения лежат в диапазоне [0, 255]), то есть по сути с градацией серого 0 – черный, 255 – белый.

Было предложено использовать «критическое» значение цвета (градации серого) для цифры *Yзд=*[0, 255]по которому будет происходить отсечка фона и выделение только той области изображения, где расположена цифра. Значение *Yзд* задается в зависимости от предполагаемого цвета цифр и фона. Если фон темнее цифр, то отсечка области изображения с фоном будет происходить при *Y<Yзд*, фон светлее цифр – *Y>Yзд*, где *Y* – значение считываемого пикселя.

С учётом вышесказанного, разработан алгоритм выделения цифры на монохромном изображении(рис. 4.1):

1) Задаётся значение *Yзд* в диапазоне [0, 255], что соответствует градации серого;

2) изображение попиксельно считывается по горизонтали и вертикали;

3) запоминаются «крайние» (слева и справа, вверху и внизу) пиксели, значение которых *Y*>*Yзд* – фон темнее цифр;

4) обрезка изображения по полученным крайним точкам – получение изображения только с цифрой.

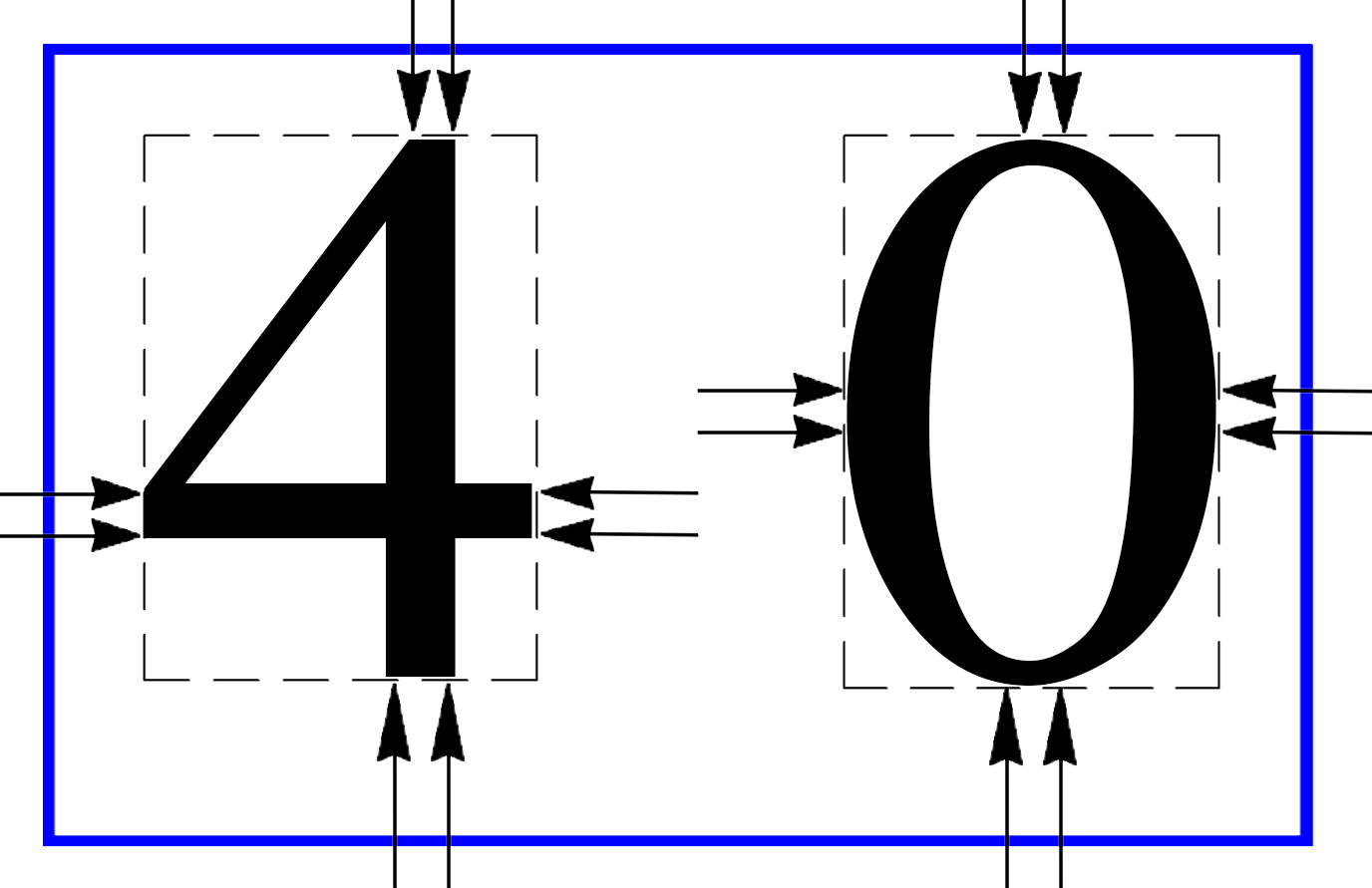


Рисунок 4.1 – Выделение цифры на фоне

3.3 Медианная фильтрация

В программе-классификаторе реализована возможность получения изображения для идентификации чисел с внешнего устройства – веб-камеры. Проблема такого изображения заключается в том, что невозможно предугадать качество освещения при котором был сделан снимок, блики на снимке и прочие дефекты. Все это при дальнейшей конвертации изображения в монохромное может привести к тому, что оно будет слишком темным или светлым. То есть заданное по умолчанию значение *Yзд* по которому выделяется область с цифрой будет неактуально и программа не сможет выделить, следовательно и распознать, цифру.

Для решения этой проблемы было предложено использовать медианную фильтрацию изображения[7]. Происходит последовательная обработка каждой точки изображения, в результате чего образуется последовательность оценок. Эта фильтрация позволяет исключить выбросы, то есть значения пикселей, которые «сильно» отличаются от остальных. При этом эти выбросы заменяются соседними значениями той же последовательности. Количество вовлекаемых в рассмотрение значений исходных пикселей называется апертурой фильтра. При апертуре *2m+1* можно исключить *m* – подряд идущих выбросов.

При медианной фильтрации используется двумерное окно (апертура фильтра – рис. 4.2), обычно имеющее центральную симметрию, при этом его центр располагается в текущей точке фильтрации. Точки изображения, оказавшиеся в пределах окна, образуют рабочую выборку текущего шага.

Обозначим рабочую выборку в виде одномерного массива ; число его элементов равняется размеру окна, а их расположение произвольно. Если упорядочить последовательность  по возрастанию, то ее медианой будет тот элемент выборки, который занимает центральное положение в этой упорядоченной последовательности. Полученное таким образом число является продуктом фильтрации для текущей точки изображения. Медианную фильтрацию можно представить в виде формулы:

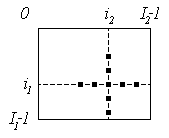


Рисунок 4.2 – Пример окна при медианной фильтрации

3.4 Разделение изображения цифры на области

Алгоритм распознавания должен быть универсален – способный работать с цифрами разных шрифтов и размеров. Нужно учитывать, что каждое изображение с идентифицируемой цифрой может иметь свое, отличное от используемого в обучаемой выборке, разрешение. Для решения этой задачи было предложено разделять изображение с цифрой на прямоугольные области(количество задается программно) и вычислять концентрацию пикселей определенного цвета в каждой из областей, что будет входным вектором параметров для ННС NEFCLASSM (Приложение 4).

Значение(количество пикселей) ширины *w* и высоты *h* обрабатываемого изображения может быть не кратно заданному количеству разбиваемых частей *n* и *m,* следовательно при вычислении ширины и высоты для областей мы получим значения вещественного типа, что недопустимо при работе с пикселями. Поэтому в таком случае при делении изображения на *n* по вертикали и *m* по горизонтали областей для одной *«крайней»* области ширина и высота будут отличаться от остальных.

Разбиение изображения(рис. 4.3) на *n*x*m* области происходит по следующему алгоритму:

1) Вычисляется ширина *w* и высота *h* полученного изображения обрезанной по краям цифры;

2) вычисляется ширина *wt* и высота *ht* прямоугольной области на которые будет делиться изображение по формуле:

3) если ширина *w* и высота *h* не кратна количеству разбиваемых частей *n* и *m*, то *wt* и *ht* округляют к меньшему – это ширина и высота для всех областей, кроме *«крайних»*.

4) вычисляется окончательная ширина и высота для *«крайних»* областей по формуле:

где *wct* – округлённое к меньшему значение *wt*, *hct* – округлённое к меньшему значение *ht*.

5) вычисляется площадь для каждой области, которая в дальнейшем используется для дальнейшего нахождения процента пикселей определенного цвета градации серого в области:

где *sc, scw, sch* – площадь «крайних» областей, *st* – площадь остальных областей.

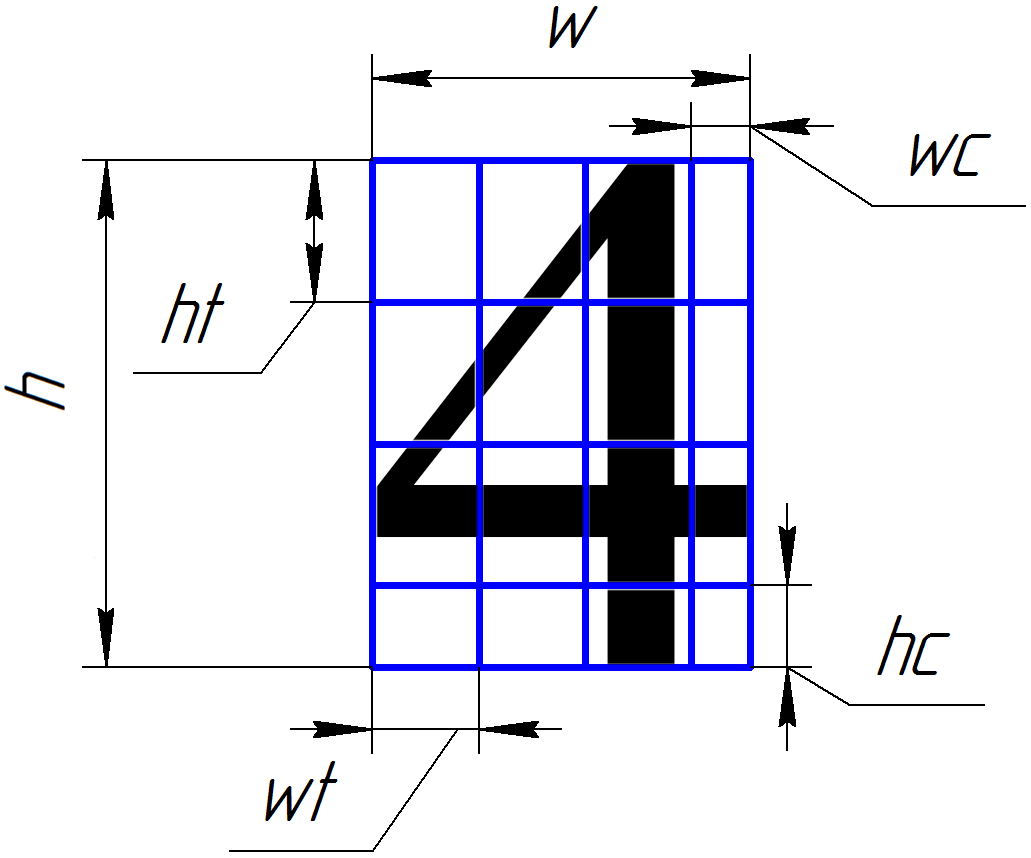


Рисунок 4.3 – Разделение изображения цифры на области

4 ОПИСАНИЕ ПРОГРАММЫ

4.1 Алгоритм работы программы

**Генерация базы правил** для ННС NEFCLASSM (блок-схема приведена в приложении 3) осуществляется с использованием обучающей выборке *L* (Приложение 2), которая содержит образцовые изображения. В качестве лингвистической переменной выступает «степень наполненности цветом», термы – лингвистические оценки «0 - пустой», «1 - почти пустой», «2 - наполовину заполнен», «3 - почти заполнен», «4 - заполнен», которые представляют собой терм-множество.

База правил генерируется по следующему алгоритму:

1) изображение с цифрой делится на *n*х*m* области;

2) для каждой области вычисляется процент концентрации пикселей, значение которых принадлежит заданному диапазону градаций серого [0, 255];

3) в зависимости от процента концентрации определяется терм из заданного терм-множества;

4) формируется правило *R* с значениями терма для каждого сегмента;

5) запись полученного правила R в базу данных.

Идентификация и распознавание числа состоит из двух этапов (блок-схема приведена в приложении 3). **Первым этапом** идентификации числа выступает определение и выделение маркировки на полученном изображение:

1) получение изображения с числом;

2) конвертация полученного цветного изображения в монохромное;

3) медианная фильтрация изображения;

4) выделение области с числом.

**Вторым этапом** идентификации числа выступает распознавание каждой цифры на полученном изображении:

1) изображение с цифрой делится на *n*х*m* области;

2) для каждой области вычисляется процент концентрации пикселей, значение которых принадлежит заданному диапазону градаций серого [0, 255];

3) в зависимости от процента концентрации определяется терм из заданного терм-множества;

4) формируется входной вектор с значениями терма для каждого сегмента и подается на вход системы классификации NEFCLASSM;

5) с помощью системы нечеткой классификации NEFCLASSM производится идентификация цифры;

6) вывод идентифицированного числа в базу данных.

4.2 Работа с программой

Для работы программы нужно получить изображение нажав «Загрузка из файла» или «Загрузка c камеры», затем нажать «Распознать». Программа выведет идентифицируемое число в текстовое поле и запишет его в файл «result.txt».

**Главное меню.**

«Файл»–>«Открыть…»–>«Файл» – загрузка изображения маркировки для идентификации из графического файла формата BMP.

«Файл»–>«Открыть…»–>«Камера» – получение изображения маркировки для идентификации с камеры.

«Файл»–>«Открыть…»–>«Сохранить как…» – сохранение текущего изображения.

«Настройки»–>«Режим работы» –>«Default» – идентификация в автоматическом режиме.

«Настройки»–>«Режим работы» –>«Debug» – идентификация в ручном режиме.

«Настройки»–>«Яркость белого» – задание образцового цвета Yзд.

«Настройки»–>«База правил» – выбор базы правил для конкретного шрифта.

«Справка»–>«ФП» – график функции принадлежности.

«Справка»–>«Помощь» – справка по работе с программой.

«Справка»–>«О программе» – информация о программе.

Кнопка «Загрузка из файла» – загрузка изображения маркировки для идентификации из графического файла формата BMP.

Кнопка «Загрузка c камеры» – получение изображения маркировки для идентификации с камеры.

Кнопка «Распознать» – идентификация числа на текущем изображении.

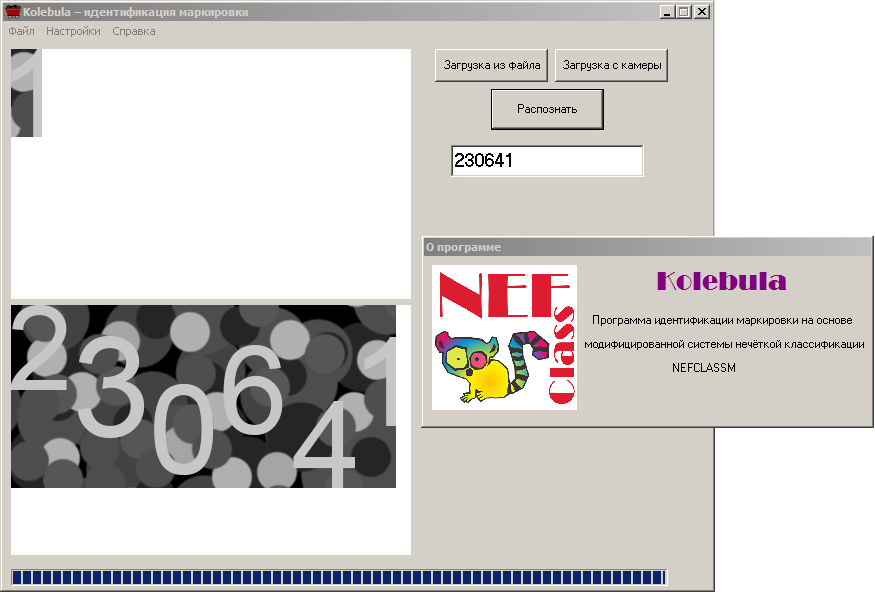
Общий вид программы-классификатора приведен на рис. 5.1

Рисунок 5.1 – Общий вид программы-классификатора

5 РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ

5.1 База правил

В результате работы программы были сгенерированы базы правил для распознавания цифр (Приложение 1). Предполагается, что может быть известно к какому конкретному типу/семейству шрифтов (например, Times New Roman) принадлежат числа, которые будет распознавать программа. В таком случае пользователь может выбрать в настройках программы базу правил только для этого шрифта – для повышения точности и скорости идентификации. Если же точная принадлежность к какому-либо шрифту распознаваемых чисел неизвестна, тогда выбирается полная база правил для нескольких шрифтов, которая также содержит правила для цифр с дефектами.

5.2 Распознавание

Были проведены эксперименты с различным разбиением цифры на области и посчитана ошибка идентификации для каждого эксперимента. Зависимость ошибки идентификации от количества разбиений приведены в таблице 6.1 и показаны на графике 6.1.

Таблица 6.1 – Зависимость ошибки идентификации от разбиений.

|  |  |  |
| --- | --- | --- |
| № | Разбиение на области | Ошибка идентификации |
| 1 | 21 область (3х7) | 0,16 |
| 2 | 25 области (5х5) | 0,28 |
| 3 | 32 области (4х8) | 0,2 |
| 4 | 32 области (8х4) | 0,25 |

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Зайченко Ю.П. Нечёткие модели и методы в интеллектуальных системах. Учебное пособие для студентов высших учебных заведений. – К.: «Издательский Дом «Слово»», 2008. – с. 344

2. Штовба С.Д. Проектирование нечетких систем средствами MATLAB . – М.: Горячая линия – Телеком, 2007. – 288 с.

3. Круглов В.В. Принятие решений на основе нечетких моделей : примеры использования. / В.В. Круглов, В.В. Борисов // – М. : Горячая линия Телеком. – 2001.

4. Банди Б., Методы оптимизации. Вводный курс: Пер. с англ. – М.: Радио и связь, 1988. – 128 с.: ил.

5. Андриевская Н. В. Моделирование систем: учеб. пособие / Н. В. Андриевская, С. В. Бочкарев. // – Пермь, 2008. – 283 с.: ил.

6. Малыхина М.П. Аспекты практического применения цветового различия для распознавания и выделения границ изображений / М.П. Малыхина, Д.А. Шичкин // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. 2013. Т. 89. № 89-89 (09). С. 676-688.

7. Грузман И.С. Цифровая обработка изображений в информационных системах: Учебное пособие. / И.С. Грузман, В.С. Киричук, В.П. Косых, Г.И. Перетягин, А.А. Спектор // – Новосибисрк: Изд-во НГТУ, 2002. - 352 c.

БЛОК-СХЕМЫ

Блок-схема генерации правил



Блок-схема алгоритма работы программы

