

С.С. САДЫКОВ,
Я.Ю. КУЛЬКОВ, С.А. РОМАНОВ,
А.С. КОРСАКОВ

**Исследование алгоритма
распознавания отдельных плоских
объектов в наложенном состоянии
на основе цепных кодов**

УДК 004.93

Муромский институт
(филиал) ФГБОУ ВО
«Владимирский
государственный
университет имени
А.Г. и Н.Г. Столетовых»,
г. Муром

В статье приведены результаты использования цепных кодов, вычисляемых по контуру бинарного изображения тестовых и реальных плоских объектов для их распознавания в наложенном состоянии в системах машинного зрения. Из цепного кода эталона наложенного объекта вычитается цепной код эталона отдельного, результат записывается, далее происходит сдвиг на одно значение по цепному коду эталона и снова производится вычитание. То же самое делается со всеми цепными кодами отдельных объектов. После чего выбирается единственный эталон, дающий минимальное значение в результате вычитания, данный эталон сохраняется в список эталонов, отобранных для распознавания. Данная операция проводится для всех наложенных объектов. На следующем этапе на каждом контуре наложенного объекта отмечается его часть, принадлежащая одному из входящих в него объектов, путем вычисления минимальной разницы между цепными кодами эталонов, при этом эталоны берутся из массива цепных кодов, отобранных на этапе обучения.

В статье приведены результаты экспериментальных исследований по распознаванию отдельных плоских объектов (ОПО) с использованием векторов цепных кодов, полученных по оболочкам бинарных изображений этих объектов.

Распознавание объектов по их изображениям – традиционная область цифровой обработки изображений. Распознавание имеет своей целью отнесение объекта к одному из заранее определенных типов. Для решения данной задачи необходимо получение вектора цепного кода объекта по его изображению, который позволит идентифицировать его с минимальной ошибкой. Для установления связи между значениями цепного кода объекта и решения о принадлежности его к определенному классу необходимо использование обучающей совокупности известных объектов [2-14].

Технология проведения экспериментов состоит из следующих основных шагов: обработка изображения, получение контура изображения объекта; вычисления цепного кода; обучение системы распознавания. Общая блок-схема алгоритма проведения экспериментов приведена на рисунке 1.

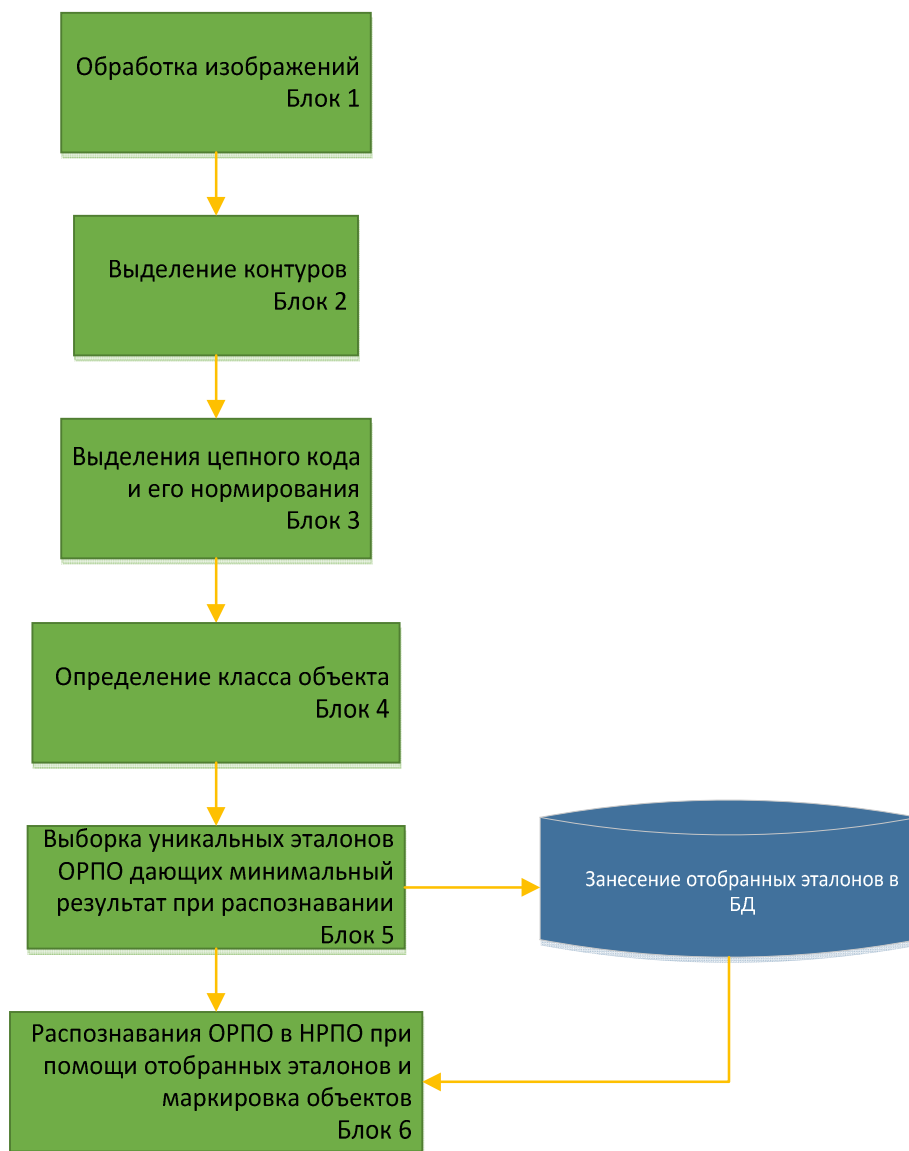


Рис. 1. Общая блок-схема технологии проведения эксперимента

Блок 1 производит подготовку изображений к дальнейшей работе с ним, в обработку входит фильтрация изображения методом Гаусса[1], бинаризация методом Оцу[2], удаление фона производится умножением текущего изображения на изображение полученное в результате бинаризации методом Оцу. Примеры исходных изображений, используемых в исследовании, представлены на рисунке 2. Первый эксперимент проводится на тестовых изображениях.

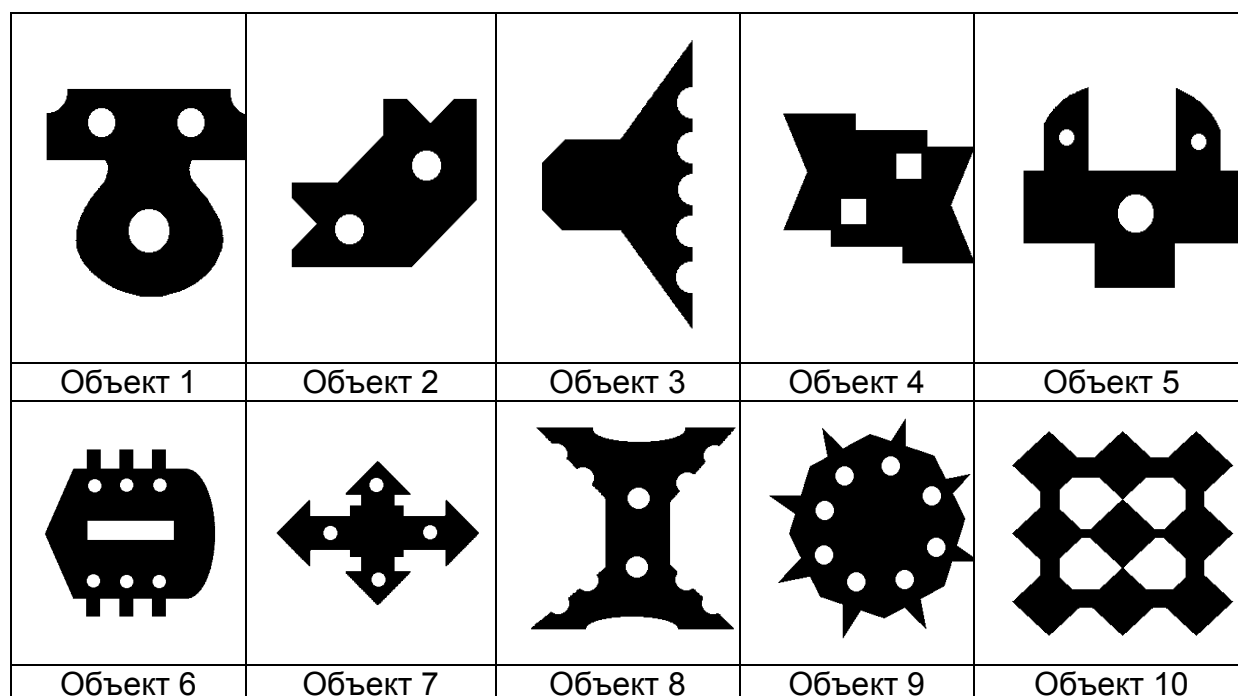


Рис. 2. Изображения реальных плоских объектов

В эксперименте участвуют наложенные изображения. Они формируются из отдельных объектов. Для формирования НТПО необходимо выбрать два случайных эталона, допускаются что эталоны могут принадлежать как одному, так и двум разным объектам. Выбранные эталоны необходимо повернуть на случайный градус, после чего наложить их на одно изображение, таким образом, чтобы они формировали один непрерывный объект. Точка, в которую будет вставлено изображение проверяется случайным образом. Два наложенных эталона образуют класс НТПО, таким образом, название класса НТПО зависит от того из каких ОТПО он состоит, например, класс 1-5 получен наложением эталонов первого и пятого объектов. Примеры наложенного НТПО представлено на рис. 3. В общей сложности мы имеем 55 комбинаций (классов), стоит учесть, что комбинации должны быть уникальными, к примеру, комбинация 2-6 и 6-2 не являются уникальными, поэтому класс 6-2 не генерируется.

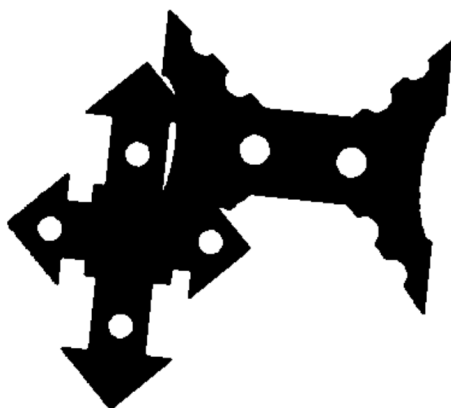


Рис. 3. Изображения наложенного тестового плоского объекта

Блок 2 формирует одноточечные контуры бинарных изображений объектов алгоритмом метода направленного перебора[3]. Примеры одноточечных контуров бинарного изображения приведены на рисунке 4.

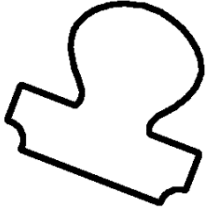


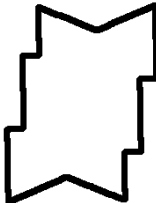
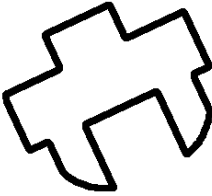
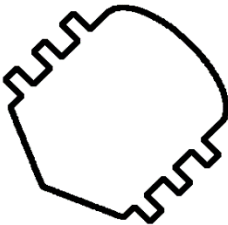

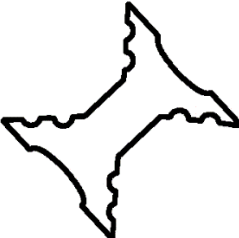
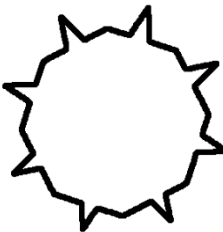
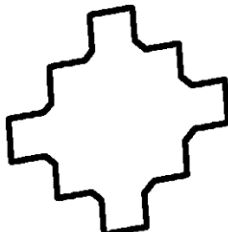
				
Объект 1	Объект 2	Объект 3	Объект 4	Объект 5
				
Объект 6	Объект 7	Объект 8	Объект 9	Объект 10

Рис. 4. Изображения контуров тестовых плоских объектов

На рисунке 5 представлен контуры наложенных изображений.

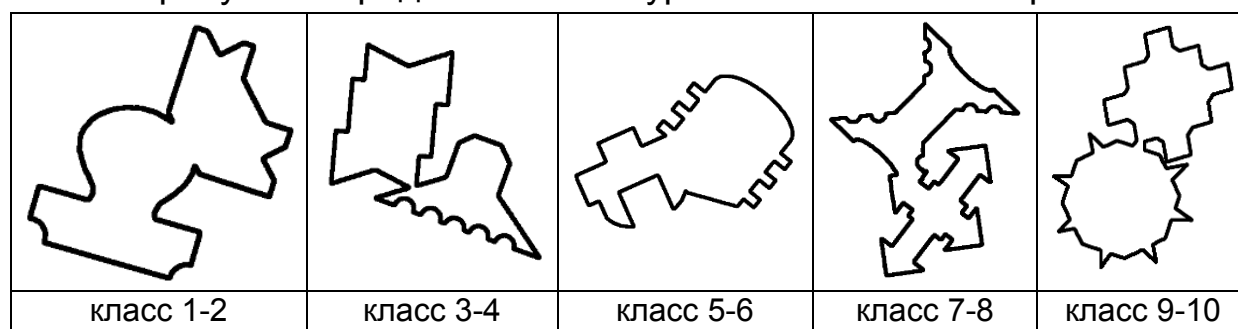


Рис.5. Изображения контуров, наложенных тестовых плоских объектов

Блок 3 Для получения односточечных безразрывных контуров бинарных изображений плоских объектов используется алгоритм направленного перебора, формирующий дискретную контурную линию из 4-х и 8-связных точек.

Одним из методов описания контуров изображений является представление с помощью цепных кодов (chain code) при использовании которых вектор, соединяющий две соседние точки, кодируется одним символом, принадлежащим конечному множеству, по методу Фримена [4]. Обычно при пользовании цепным кодом рассматривается окрестность точки размером 3×3 и 8 возможных направлений кодирования. В основе этого представления лежит 8-связная решётка (рис.6).

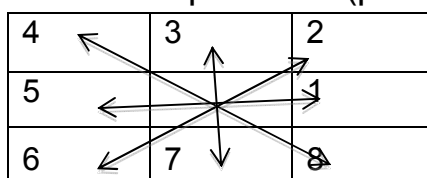
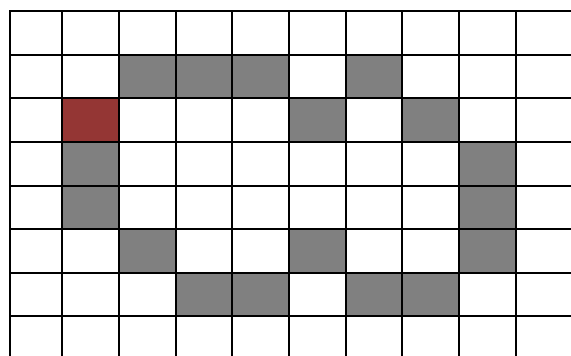


Рис. 6 Схема обозначений возможных направлений перемещения от текущей точки на дискретном изображении

Начиная с первой точки, производится обход контура по часовой стрелке, при этом каждая последующая точка кодируется числом 1-8, в зависимости от своего расположения относительно центральной точки окрестности. Результатом кодирования является последовательность, состоящая из цифр 1-8.

На рис. 7а приведен пример контура и его цепной код (рис.7б)



[2 1 1 8 2 8 8 7 7 6 5 4 6 5 4 4 3 3]
б)

а)

Рис. 7. Пример контура и его представление в виде цепного кода (красным выделена начальная точка сканирования)

Полученные значения цепного кода нормируются ($K_n = K_k / 8$) и заносятся в таблицу 1.

Результаты определения значений цепного кода контура

Таблица 1

№ п/п точки контура	Объект 1	Объект 2	...	Объект 2000
1	0,25	0,25	...	0,5
2	0,125	0,75	...	1
...
M	0,25	0,375	...	1

Содержание каждого столбца таблицы 1 представляет собой множество значений представления контура конкретного объекта в виде цепного кода.

Блок 4 в данном блоке происходит распознавание класса объекта методом Ближайших соседей[5].

Блок 5 После установления класса НТПО, необходимо определить месторасположение каждого ОТПО образующих данный сложный объект.

Для этого функция цепного кода НТПО, класс которого установлен, сравнивается со всеми эталонными K_3 -функциями двух

ОТПО, образующих данный НТПО, в нашем случае это будут эталонные α_3 -функции ОТПО 1 и ОТПО 2, по формулам:

$$\begin{aligned}\Delta_1 &= \frac{1}{k_1} \sum_{i=1}^{k_1} |\alpha_{эj} - \alpha_{сли}| \\ \Delta_2 &= \frac{1}{k_1} \sum_{i=2}^{k_1+1} |\alpha_{эj} - \alpha_{сли}|, \\ &\dots \\ \Delta_m &= \frac{1}{k_1} \sum_{i=m}^{k_1+m} |\alpha_{эj} - \alpha_{сли}| \end{aligned} \quad (1)$$

где

$K_{эj}$ – значения элементов K_3 -функции эталонного ОТПО ($j=1, 2, \dots, N$);

$K_{сли}$ – значения элементов K_3 -функции эталонного сложного объекта НТПО O_i ($i=1, 2, \dots, P$);

m – число смещений α_3 -функции эталонного ОТПО $K_{эj}$ для полного обхода контура сложного объекта НТПО O_i ;

Значение m определяется как:

$$m = 2(P - N) \quad (2)$$

где

N – число точек контура эталонного ОТПО;

P – число точек контура сложного объекта НТПО O_i .

K_3 -функция эталонного ОТПО сравнивается с удвоенной K_3 -функцией эталонного сложного объекта НТПО.

Решение об идентификации принимается на основе следующего выражения

$$\min R = \min_{i=1,q} \{R_i\}, \quad (3)$$

где q – число эталонов.

Аналогичное сравнение всех α -функций ОТПО 2 выполняется с K_3 -функцией найденного НТПО.

Блок-схема работы алгоритма представлена на рисунке 8

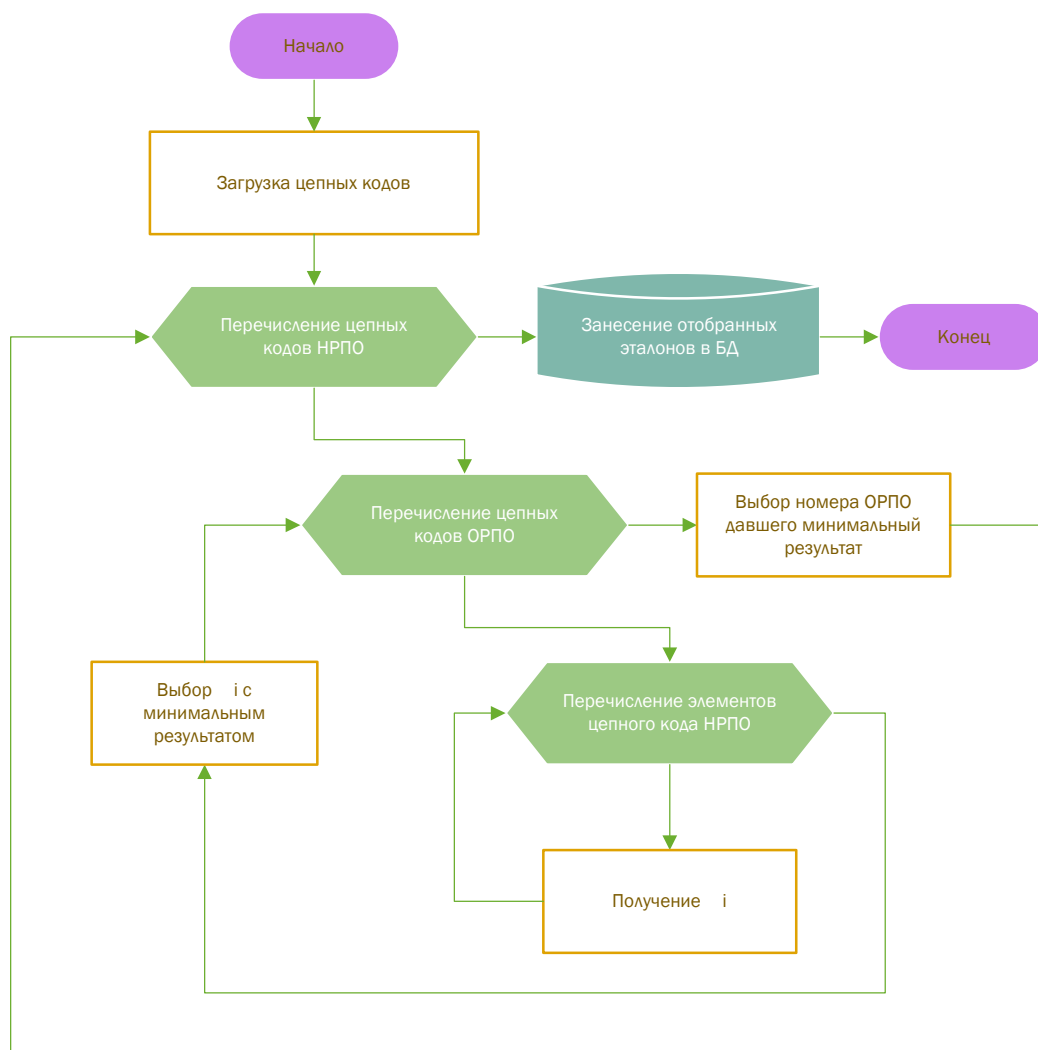


Рис. 8. Алгоритм обучения

Отобранные эталоны сохраняются, количество эталонов, отобранное для распознавания каждого класса представлено на таблице 2.

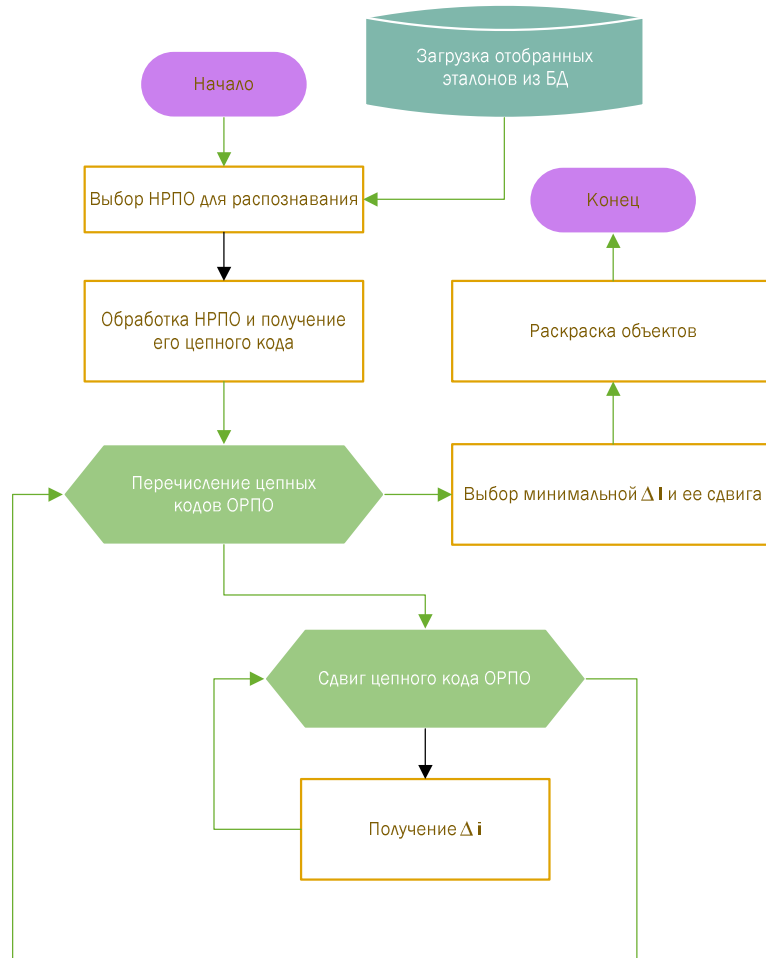
Таблица 2

Отобранные эталоны.

Класс		Этапонов 1	Этапонов 2
1	1+1	280	286
2	1+2	272	283
3	1+3	270	281
4	1+4	271	284
5	1+5	272	282
6	1+6	269	283
7	1+7	276	280
8	1+8	277	285
9	1+9	278	279
10	1+10	273	282
11	2+2	291	292
12	2+3	284	289
13	2+4	284	273

14	2+5	287	289
15	2+6	290	280
16	2+7	286	284
17	2+8	287	282
18	2+9	282	285
19	2+10	283	286
20	3+3	292	289
21	3+4	282	280
22	3+5	283	282
23	3+6	282	286
24	3+7	288	283
25	3+8	289	284
26	3+9	284	287
27	3+10	290	280
28	4+4	277	291
29	4+5	274	284
30	4+6	273	281
31	4+7	275	286
32	4+8	270	290
33	4+9	269	290
34	4+10	269	285
35	5+5	283	290
36	5+6	280	277
37	5+7	277	283
38	5+8	276	288
39	5+9	280	285
40	5+10	281	284
41	6+6	275	294
42	6+7	271	282
43	6+8	272	288
44	6+9	277	290
45	6+10	274	285
46	7+7	284	292
47	7+8	281	290
48	7+9	279	288
49	7+10	271	289
50	8+8	290	292
51	8+9	285	288
52	8+10	288	284
53	9+9	285	290
54	9+10	282	280
55	10+10	278	289

Блок 6 На данном этапе приложение выделяет цветом границы двух ОТПО. Процесс протекает следующим образом, по формулам 1,2 и 3 мы выбираем два минимальных значения и запоминаем цепные коды и сдвиги, которые позволили получить минимальную . Затем мы производим сдвиг сохраненного цепного кода на указанное значение и вычитаем это значение из цепного кода НТПО. Участки, давшие значение 0 окрашиваем в цвет соответствующего ОТПО. Соответствующие действия проводим для второго цепного кода. Блок-схема, отображающая работу данного алгоритма, представлена на рисунке 9.



Данный метод позволяет маркировать положение всех ОТПО в НТПО, поступивших в программу изображений. Результат маркировки представлен на рисунке 10.

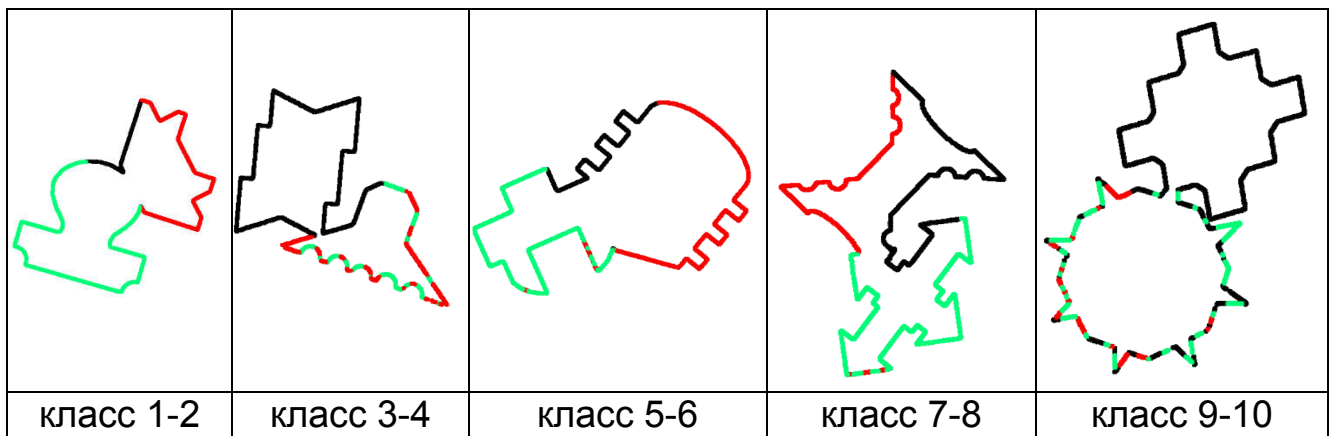


Рис. 10. Результат раскраски объекта.

Для раскраски одного объекта, без учета распознавания класса и загрузки отобранных эталонов в память программы, в зависимости

от количества эталонов программой затрачивается время в соответствии с рисунком 11

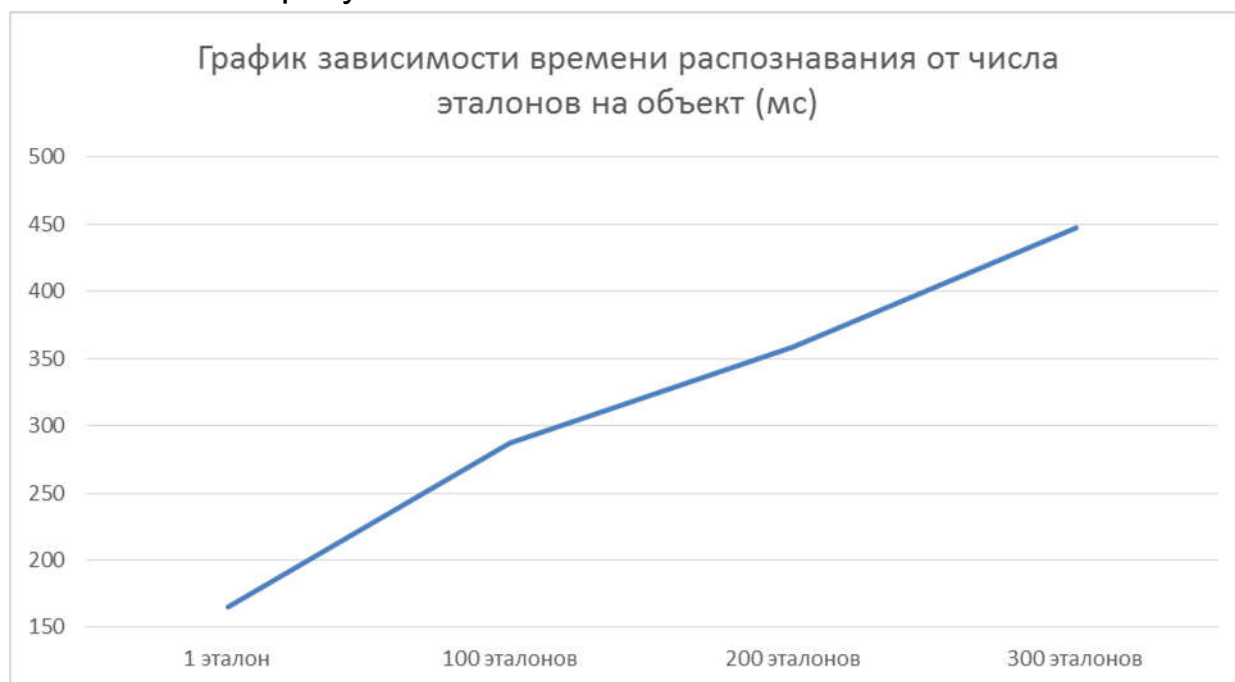


Рис. 11. График зависимости времени распознавания от числа эталонов на объект

Эксперимент на реальных плоских объектах (РПО)

После получения результатов обучения данным методам на тестовых изображениях проведен эксперимент над реальными объектами. Отдельные реальные плоские объекты (ОРПО) представлены на рисунке 12.

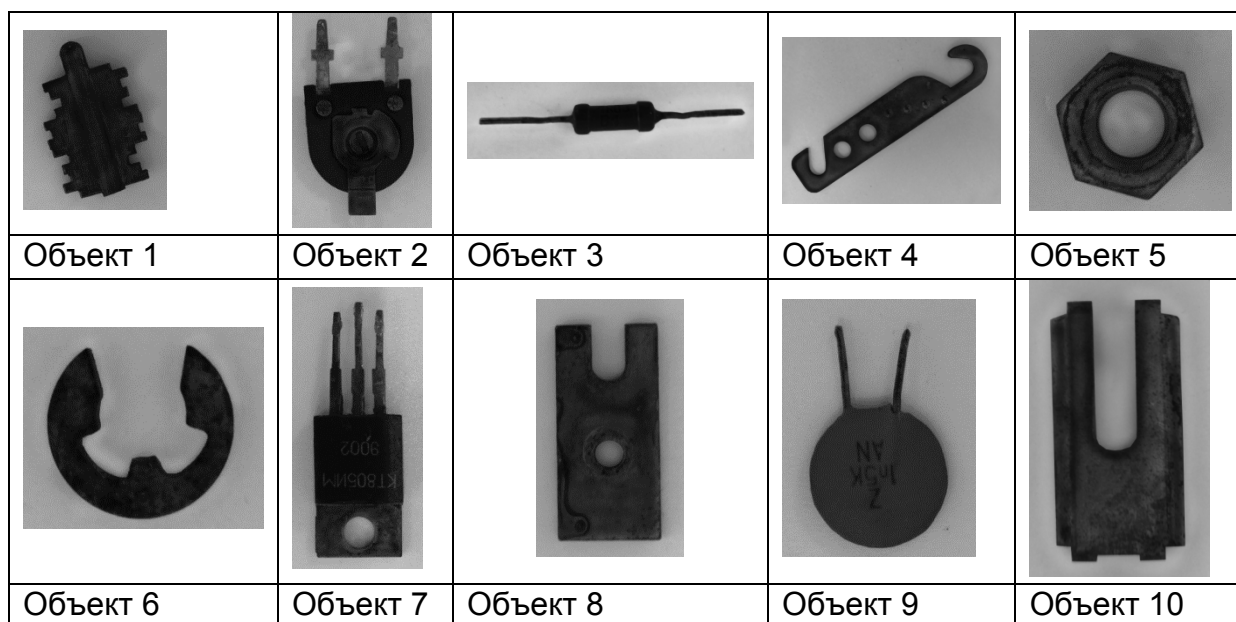


Рис. 12. Отдельные реальные плоские объекты.

При помощи алгоритма, описанного в блоке 1 формируются наложенные реальные плоские объекты (НРПО) их примеры представлены на рисунке 13.

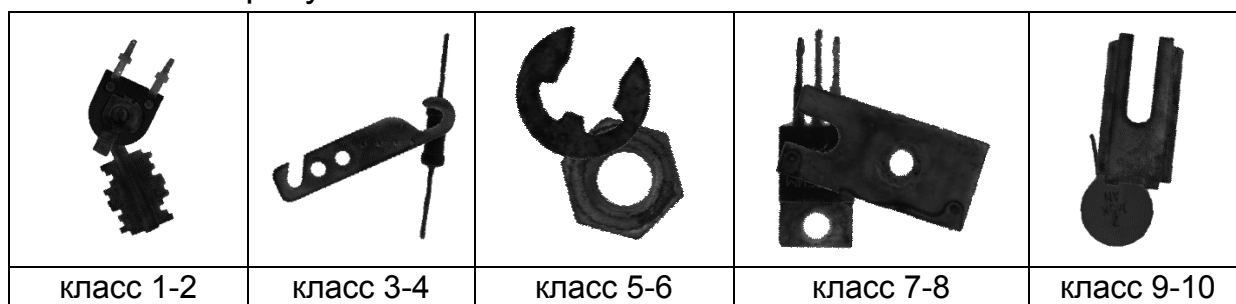


Рис. 13. Наложённые реальные плоские объекты.

Получив все НРПО, выделяются контуры ОРПО, согласно алгоритмам блока 2, формируются одноточечные контуры бинарных изображений объектов алгоритмом метода направленного перебора. Примеры одноточечных контуров бинарного изображения ОРПО приведены на рисунке 14.

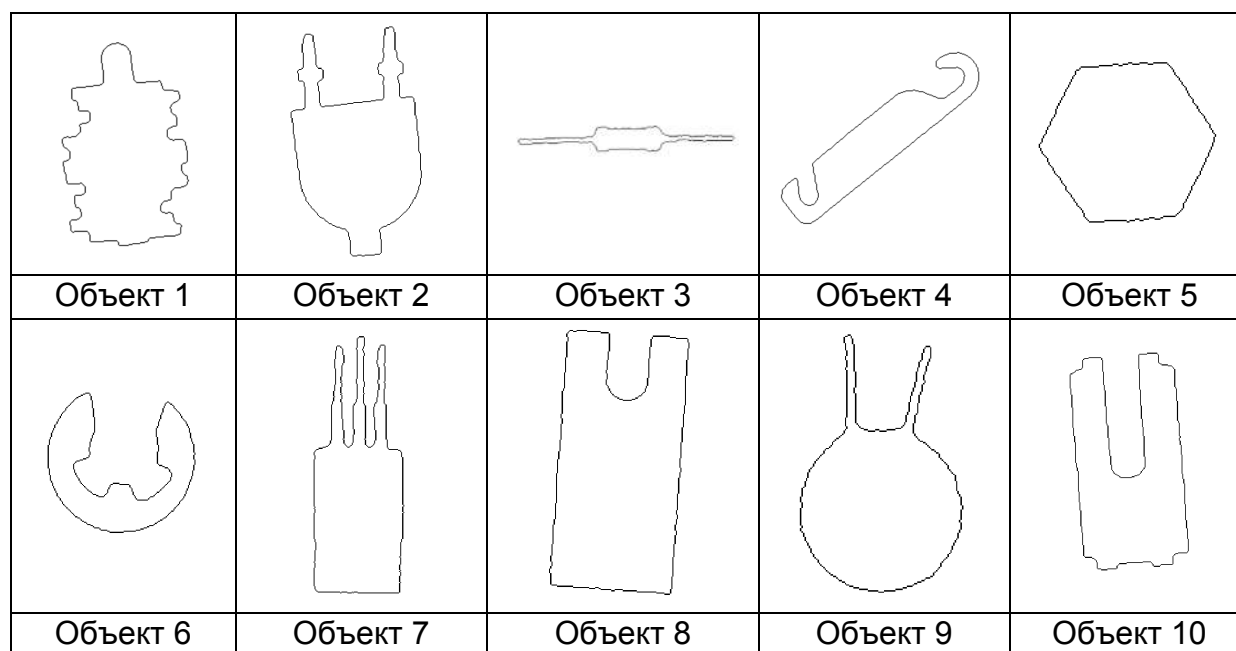


Рис. 14. Изображения контуров реальных плоских объектов

На рисунке 15 представлены контуры наложенных изображений.

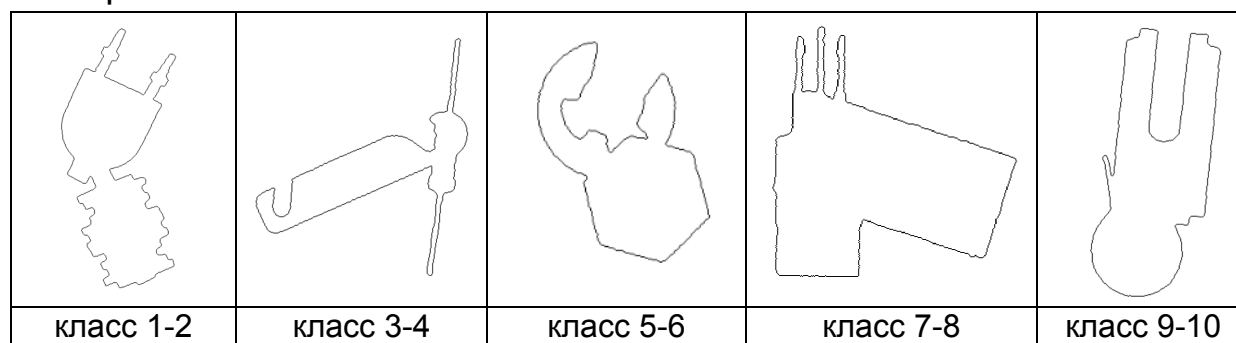


Рис.15. Изображения контуров, наложенных реальных плоских объектов

Затем с полученными контурами производится обучение по алгоритмам представленных в **блоках 3-6**, в результате чего получается таблица отобранных эталонов необходимых для распознавания всех НРПО, данные о количестве эталонов необходимых для распознавания представлены в таблице 3.

Таблица 3

Отобранные эталоны.

Класс		Эталонов 1	Эталонов 2
1	1+1	280	
2	1+2	272	277
3	1+3	270	271
4	1+4	271	272

5	1+5	274	249
6	1+6	271	228
7	1+7	275	234
8	1+8	273	265
9	1+9	276	284
10	1+10	276	262
11	2+2	279	
12	2+3	270	272
13	2+4	273	272
14	2+5	270	270
15	2+6	273	252
16	2+7	275	234
17	2+8	269	242
18	2+9	278	263
19	2+10	279	280
20	3+3	276	
21	3+4	274	280
22	3+5	268	270
23	3+6	276	247
24	3+7	274	229
25	3+8	273	236
26	3+9	276	260
27	3+10	274	285
28	4+4	270	
29	4+5	268	245
30	4+6	266	235
31	4+7	265	234
32	4+8	267	265
33	4+9	272	281
34	4+10	261	266
35	5+5	251	
36	5+6	245	231
37	5+7	252	243
38	5+8	241	262
39	5+9	249	279
40	5+10	245	258
41	6+6	233	
42	6+7	225	242
43	6+8	235	263
44	6+9	229	281
45	6+10	235	261
46	7+7	242	
47	7+8	240	261
48	7+9	246	283
49	7+10	243	268
50	8+8	264	
51	8+9	254	282
52	8+10	260	268
53	9+9	283	
54	9+10	275	263
55	10+10	266	

По алгоритму **блока 6** проводится экзамен, результаты маркировки которого представлены на рисунке 16.

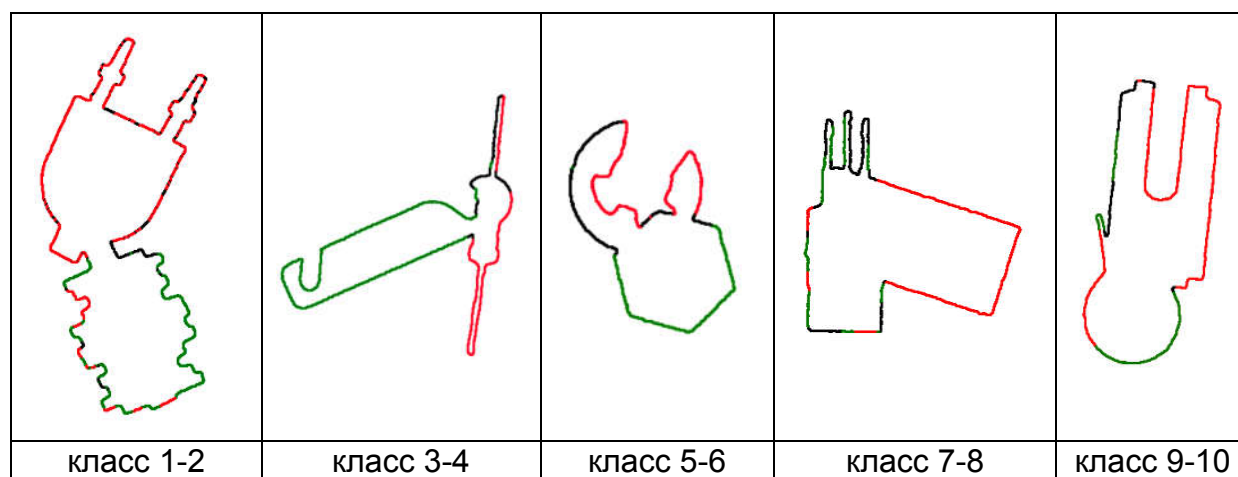


Рис.16. Результат раскраски объекта.

Заключение

1. Эксперимент показал, что наиболее удачными при распознавании являются либо классы, состоящие из двух эталонов одного ОТПО или два ОТПО имеющих наиболее отличную друг от друга форму. В частности объекты, для наилучшего распознавания, не должны иметь длинные прямые участки и большие скругления.

2. В процессе проведения работы было установлено, в наложенных объектах, у изображений которых одна из деталей делит вторую на две части, алгоритм позволяет маркировать лишь большую часть отдельного объекта, находящуюся по одну сторону от пересекаемой детали, меньшая же часть остается без маркировки.

3. Большое влияние на распознавание оказывает способы обработки. Таким образом порог бинаризации и фильтрация должны быть одинаковыми у ОТПО и НТПО, так как в ином случае появляются неточности в виде добавления дополнительных пикселей изображения за счет разного порога бинаризации, что вносит значительные изменения в цепной код изображения.

4. В процессе проведения эксперимента, по определению класса при помощи цепных кодов, было выявлено, что такое распознавание дает очень плохие результаты, из чего можно сделать вывод, что цепные коды можно применить только в том случае если нам уже известен класс наложенного объекта.

5. Большое влияние на распознавание объекта оказывает процент наложения, чем выше процент тем меньше вероятность

распознавания, так как высокий процент наложения делает невозможным сохранение последовательности цепного кода ОТПО из которых состоит объект.

Литература

1. Н. Ш. Кремер, 2.3. «Метод Гаусса»;
2. N. Otsu (1979). «A threshold selection method from gray-level histograms».
3. Савченко А.В. «Метод направленного перебора альтернатив в задачах распознавания образов»; № 11 (77), 11.2011
4. Н.М. Новикова «СТРУКТУРНОЕ РАСПОЗНАВАНИЕ ОБРАЗОВ» Издательско-полиграфический центр Воронежского государственного университета 2008
5. Sadykov S.S., Kulkov Y.Yu. Research of recognition of the imposed flat objects on dimensionless marks of their contours // Pattern Recognition and Information Processing: Proc. of 13th Intern. Conf. (3-6 Oct. 2016, Minsk, Belarus) / ed.: S. Ablameyko, V. Krasnoproshin. – Minsk: Publ. Center of BSU, 2016, pp.205-208