

Министерство образования и науки Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Алтайский государственный технический университет им. И.И. Ползунова»

Факультет (институт) Информационных технологий

Кафедра Прикладная математика

Направление Программная инженерия

УДК 004.9

Допустить к защите в ГЭК

Заведующий кафедрой

С. А. Кантор

(подпись)

(инициалы, фамилия)

« 28 » март 2016 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

МД 09.04.04.04.000 ПЗ

(обозначение документа)

**Разработка системы автоматического распознавания дорожной
обстановки в видеопотоке**
(тема магистерской диссертации)

Пояснительная записка

Студент группы 8ПИ-41

Уваров Д. А.

(фамилия, имя, отчество)

Научный руководитель профессор, к.ф.-м.н.

Крючкова Е.Н.

(должность, ученая степень)

(инициалы, фамилия)

Научные консультанты:

(должность, ученая степень) (подпись) (инициалы, фамилия)

(должность, ученая степень) (подпись) (инициалы, фамилия)

Барнаул 2016

Реферат

Данная работа посвящена разработке системы компьютерного зрения для распознавания знаков дорожного движения. Разработанная программное обеспечение позволяет распознать и классифицировать знаки дорожного движения в видеопотоке, записанного с автомобильного видеорегистратора.

Объём работы – 81 страница, включающих 16 рисунков, 1 таблицу, 34 используемых источников.

Ключевые слова: компьютерное зрение, классификация изображений, знаки дорожного движения.





The abstract

The purpose of the research is to develop a computer vision system for traffic sign recognition. The developed software allows to detect and classify traffic sign in video stream obtained from a car dash camera.

Size of work – 81 pages includes 16 figures, 1 table, 34 used sources.

Key words: computer vision, image classification, traffic signs.

Работа допущена к опубликованию с изъятием стр. с 49 по 81 в соответствии с п. 38 Приказа Минобрнауки от 29.06.2015г. № 636

						МД 09.04.04.04.000 ПЗ			
Изм.	Коп. уч.	Лист	№ док.	Подпись	Дата	Разработка системы автоматического распознавания дорожной обстановки в видеопотоке	Стадия	Лист	Листов
Разработал	Уваров Д. А.				29.07.16		У	2	81
Рук. проекта	Крючкова Е.Н.				23.06.16				
Н. контролер	Потупчик А. И.				27.06.16				
Зав. кафедр.	Кантор С. А.				27.06.16		АлтГТУ ФИТ 8ПИ - 41		

Содержание

Введение	5
1 Описание предметной области	6
1.1 Проблемы распознавания дорожных знаков	6
1.2 Анализ современного состояния дел	7
2 Постановка задачи и подход к ее решению	15
3 Предварительная обработка	16
3.1 Перевод в цветовое пространство $L^*a^*b^*$	16
3.2 OTSU бинаризация	18
4 Поиск симметричных областей	21
4.1 Быстрый метод поиска радиальной симметрии	21
4.2 Оптимизация	24
5 Классификация знаков дорожного движения	25
5.1 Использование цвета	25
5.2 Deskriptor интересных точек	25
5.2.1 SIFT-deskriptory	25
5.2.2 Гистограмма направленных градиентов (HOG)	28
5.3 Метод опорных векторов (SVM)	29
6 Состав и структура системы	31
6.1 Модуль распознавания	32
6.2 Модуль слежения	32
7 Описание программного обеспечения	35
7.1 Описание основных разработанных классов	35
Заключение	42

Список использованных источников	43
Приложение А Задание на магистерскую диссертацию	47
Приложение Б Исходный код	49

Введение

В современном мире автомобили занимают важную часть нашей жизни. С каждым годом на дорогах появляется все больше и больше личных машин. С ростом количества транспортных средств на дорогах общего пользования растет и количество аварий и несчастных случаев. Большая часть происшествий происходит из-за невнимательности водителей.

Для увеличения безопасности на дорогах используются множество систем. Часть из них полностью забирают управление на себя в случае экстренной ситуации, например система автоматического экстренного торможения. Часть из них только оповещает водителя, например системы автоматического распознавания знаков дорожного движения.

Главным достоинством такой системы распознавания знаков является уменьшение рисков, возлагаемых на водителя, так как увеличивается получаемая им информация.

Данная работа является пояснительной запиской к магистерской диссертации, выполненной на кафедре «Прикладная математика» Алтайского Государственного технического университета им. И. И. Ползунова. Целью данной работы является разработка системы автоматического распознавания дорожной обстановки в видеопотоке.

1 Описание предметной области

В современно мире все шире получают распространение беспилотные автомобили. В некоторых корпорация (например Google) уже сейчас проводятся эксперименты управления автотранспортом компьютерными системами, а в некоторых местах уже вводят законы, регулирующие данный вид транспорта.

В таких сложных системах, при движении по автомобильной дороге, необходимо своевременно и точно получать информацию о дорожной обстановке для безопасного и эффективного вождения. Одним из источников данной информации являются знаки дорожного движения. Таким образом, системы распознавания и классификации дорожных знаков очень важны.

Область действия данных систем не ограничиваются только беспилотными автомобилями. Уже сейчас некоторые производители легковых автомобилей внедряют системы распознавания знаков при движении. Это позволяет системе автомобиля реагировать на различную дорожную обстановку (например информировать при превышении скорости), а так же повышает безопасность и комфорт управления транспортным средством. Повсеместное использование данных интеллектуальных систем может значительно сократить количество аварий на дороге, связанных из-за невнимательности водителей.

1.1 Проблемы распознавания дорожных знаков

Так как знаки дорожного движения могут значительно отличаться между собой по форме и цвету, задача распознавания знаков является непростой задачей компьютерного зрения. К многообразию форм знаков так же добавляется необходимость определения положения знака в пространстве, так как для некоторых знаков важно расположение на дороге. Например знак «Остановка запрещена», изображенный на рисунке 1.1. Его действие распространяется на ту сторону дороги, на которой он установлен. Для других важно положение относительно других знаков (Рисунок 1.2).

Таким образом, для систем распознавания знаков дорожного движения стоит задача не только поиска и классификации знаков, а так же определение их положе-



Рисунок 1.1 — Знак «Остановка запрещена»



Рисунок 1.2 — Знак «Ограничение минимальной дистанции» с указанием дальности действия

ния относительно дороги и друг друга для получения более точной и корректной информации о дорожной обстановке.

1.2 Анализ современного состояния дел

Задача распознавания знаков дорожного движения является одной из широко изучаемых задач компьютерного зрения.

В [1] автор предложил алгоритм, способный распознавать знаки «Стоп», «Уступите дорогу» и «Проезд запрещен». Он содержит 6 модулей: цветовая сегментация, локализация границ, RGB-дифференцирование, определение границ, извлечение гистограмм и классификация. Цветовая сегментация в данной работе используется только для локализации областей красных границ.

В работе [2] представлен метод определения дорожных знаков, основанный на форме знака. В данной работе используется алгоритм для определения правильных многоугольников. На тестовой выборке описанный метод показал точность

около 95%. Данный метод инвариантен к поворотам в плоскости знака и может быть использован для приложений, работающих в реальном времени.

В работе [3] представлена нейронная сеть с входным шаблоном 18×18 точек. Шаблон подается в нейронную сеть, содержащую 324 нейронов на входном слое, 15 нейронов в скрытом слое и три нейрона в выходном слое. Система смогла распознать 23 из 24 знаков ограничения скорости и 23 из 24 других знаков.

В работах [4; 5] предлагается метод определения дорожных знаков используя только яркость. Объекты на изображении отделяются от фона используя фильтр Лапласа после сглаживания изображения для удаления шумов. Затем получается бинаризованное изображение после применения определенной пороговой обработки. Далее, для определения знаков, используется генетический алгоритм, который используется для поиска круглых шаблонов.

В работе [6] описывается метод распознавания знаков, основанный на сопоставлении с шаблонами. Данный метод может определять знаки круглой и треугольной формы. Используются различные шаблоны с радиусами от 7 до 18 пикселей. Каждый шаблон разделен на 8 частей, основанных на ориентации ребер. Данный метод успешно определяет знаки в 90% случаях.

В работе [7] рассматривается проблема распознавания знаков в условиях со слабым освещением, таких как сумерки, где информация о цвете довольно скудная. Исходное изображение обрабатывается двумя фильтрами Собеля для вычисления величины и направления градиента. Граничные точки выбираются по порогу величины градиента. Данные точки сегментируются в два шага, выстроенных иерархическим образом. Коэффициент распознавания данного метода 95% для знаков треугольной и эллиптической формы и 93% для знаков прямоугольной формы.

В [8] рассматривается алгоритм обнаружения и локализации объектов на статическом изображении, основанные на сопоставлении с эталоном. Для повышения эффективности предлагается использовать минимаксные функции схожести. В работе рассмотрены различные функции данного типа для обработки изображений. Так же рассмотрены вычислительные затраты при определении схожести двух изображений различными функциями схожести и показано, что функции ис-

следуемого класса формируют интегральную оценку схожести на основе последовательного анализа элементов обрабатываемых изображений и позволяет получить нормированный минимаксный критерий схожести.

В работе [9] рассматривается система автоматического распознавания дорожных знаков, в которой регионы интереса, содержащие дорожные знаки, определяются аффинным уточнением. В работе используются черно-белые изображения. Между целевым изображением и существующими шаблонами вычисляются нормализованные Евклидовы расстояния. Шаблон с минимальным значением расстояния выбирается в качестве подходящего (описывающего знак).

В работах [10—12] предложены методы для распознавания знаков дорожного движения из движущейся картинки. Алгоритм основан на фильтрах краев. Сначала использует обычный фильтр Лапласа для извлечения границ знаков, затем «простой векторный фильтр» (the Simple Vector Filter (SVF)) для извлечения определенного цвета из изображения. Для классификации изображения используется генетический алгоритм, ищущий круглые знаки дорожного движения.

В работе [13] представлен метод определения дорожных знаков, основанный на форме знака. В данной работе используется алгоритм для определения правильных многоугольников. На тестовой выборке описанный метод показал точность около 95%. Данный метод инвариантен к поворотам в плоскости знака и может быть использован для приложений, работающих в реальном времени.

В работе [14] рассматриваются два различных алгоритма для определения дорожных знаков. В первом уровне серого вместе с простыми геометрическими критериями используются для определения дорожных знаков. Во втором алгоритме изображение переводится в цветовое пространство HSV, из которого используются каналы цветового тона (H) и насыщенности (S). Изображение разделяется на регионы 16 точек, которые классифицируются как 0 и 1 в зависимости от количества точек, удовлетворяющих определенному ограничению. Дальнейшей обработке подвергаются только регионы, помеченные как 1. Детектор форм основан на геометрическом анализе границ контура.

В [15] рассматривается задача локализации объектов на изображении в ситуациях, когда масштаб искомого объекта на изображении несколько отличается от масштаба эталонного изображения объекта. В работе используется алгоритм, использующий процедуру голосования в накопительном пространстве (accumulator space), основанный на вычислении трансформации Хафа. Авторами рассматривается технология локализация предупреждающих дорожных знаков в системе анализа дорожной ситуации.

В работе [16] предложена система распознавания дорожных знаков, которая состоит из двух модулей:

- Модуль определения знаков на изображении.
- Модуль классификации найденных знаков.

Метод определения основан на цвете и форме знака. RGB изображения конвертируется в пространство цветов HSV, которое разделяется на несколько регионов. Каналы S и V используются для поиска регионов с определенным тоном. Бинарное изображение, полученное на первом шаге, используется для извлечения формы знака основываясь на знаниях о дорожных знаках. Изображение сравнивается с внутренней базой, содержащей различные образцы форм знаков и выбирается образец с минимальным отличием.

В работе [17] используется простая нейронная сеть для определения цвета и формы дорожного знака. Оригинальное изображение сперва обрабатывается фильтром Лапласиана-Гауссиана. После определения формы и цвета изображения используется сравнение с образцами для конечного распознавания.

В работе [18] рассматривается система автоматического распознавания дорожных знаков, в которой регионы интереса, содержащие дорожные знаки, определяются аффинным уточнением. В работе используются черно-белые изображения. Между целевым изображением и существующими шаблонами вычисляются нормализованные Евклидовы расстояния. Шаблон с минимальным значением расстояния выбирается в качестве подходящего (описывающего знак).

В работе [19] представлена система распознавания знаков превышения скорости. Сперва определяются красные области на изображении. Затем используется

многослойная нейронная сеть для определения позиции и размеров окружности знака. Для классификации символов, изображенных на знаке, так же используется нейронная сеть. В данной работе достигнут коэффициент распознавания 98.3%.

В работе [20] используется нейронная сеть для распознавания предупреждающих знаков. На вход нейронной сети поступает прямоугольник 10, представляющий желтую область внутри предупреждающего знака. Нейронная сеть имеет 100 нейронов на входном слое. Выходной слой содержит два выхода: «знак» и «не знак». В скрытом слое используются 30 узлов. Система смогла определить 55% предупреждающих знаков из 55 изображений. Для больших изображений процент распознавания увеличивается до 86%.

В работе [21] представлена нейронная сеть для распознавания, которая содержит в себе три подсети:

- 1) Нейронная сеть для классификации.
- 2) Нейронная сеть Хопфилда
- 3) Нейронная сеть для проверки.

В работе [22] представлена нейронная сеть с входным шаблоном 18×18 точек. Шаблон подается в нейронную сеть, содержащую 324 нейронов на входном слое, 15 нейронов в скрытом слое и три нейрона в выходном слое. Система смогла распознать 23 из 24 знаков ограничения скорости и 23 из 24 других знаков.

В работе [23] показана вычислительная модель локализации объектов на цифровом изображении, разработанной в предположении иерархической природы объектов посредством обобщения и развития вычислений в методах Оцу, модели Мамфорда-Шаха и методе K -средних. В работе показано, что перечисленные подходы, в отличие от большинства других подходов к выделению объектов на цифровых изображениях, наиболее формализованы по постановке задачи, так как при построении разбиений опираются на определенный критерий качества. Критерием качества данных методов служит суммарная квадратичная ошибка или равносильная величина среднеквадратичного отклонения значений пикселей изображения от значения пикселей оцениваемого кусочно-постоянного приближения. В данной работе локализация объектов сводится к получению последовательности разби-

ний изображения на вложенные изображения из условия минимизации суммарной квадратичной ошибки или среднеквадратичного отклонения.

В работах [24; 25] используются для классификации дорожных знаков используются сети адаптивного резонанса ART1.

В работе [26] используется нейронная сеть для определения позиции и размера знаков ограничения скорости. Для распознавания символов, изображенных на знаке, используется классификатор, основанный на нейронной сети. Был достигнут коэффициент определения 100% и коэффициент распознавания 98.3%.

В работе [27] используется метод опорных векторов (SVM) для классификации формы знаков. Данный метод смог распознать 82% открытых знаков и 54% загороженных знаков.

В работе [28] автор предлагает использовать кривые Гилиса для определения положения знаков на изображении.

Gielis Curves[29] (так же известна как суперформула) — это кривые, впервые представленные Йоханом Гилисом в 2003 году. Гилис предложил формулу, которая может быть использована для описания многих сложных форм и кривых в природе.

Полярные координаты данных кривых определяются по формуле (1.1):

$$r(\phi) = \left(\left| \frac{\cos\left(\frac{m\phi}{4}\right)^{n_2}}{a} \right| + \left| \frac{\sin\left(\frac{m\phi}{4}\right)^{n_3}}{b} \right| \right)^{-\frac{1}{n_1}} \quad (1.1)$$

, где r и ϕ — координата точки в полярных координатах a , b , n_1 , n_2 , n_3 , m — параметры, от которых зависит форма кривой.

Детектор дорожных знаков, использующего данные кривые, включает в себя этапы:

- 1) IHLS преобразование — преобразование изображения в пространство цветов IHLS.
- 2) По заданному ограничению бинаризируем изображение.
- 3) Удаление шумов, присутствующих на изображении, а так же мелких объектов, которые могут привести к плохим результатам детектора. После

данного шага должны остаться контуры потенциальных знаков дорожного движения.

- 4) Применение алгоритма поиска выпуклой оболочки для избавление от невыпуклых контуров.
- 5) Подбор параметров для восстановление формы кривой. На данном этапе могут быть использованы различные оптимизации для более точного и быстрого определения формы знака.

Данный детектор позволяет находить положение и форму знаков, но имеет ряд недостатков:

- 1) Находятся не только знаки дорожного движения (рисунок 1.3).
- 2) Сильное увеличение времени работы алгоритма при увеличении размера изображения.



Рисунок 1.3 — Определенный знак с найденными лишними областями

2 Постановка задачи и подход к ее решению

Имеется видео, полученное с автомобильного видеорегистратора. Целью данной работы является разработка системы для автоматического определения дорожных знаков, а так же их классификации. Обработку видео можно свести к последовательной обработке кадров с учетом информации, полученной на предыдущих шагах. Так как перемещение знаков происходит непрерывно, то при обработке очередного кадра существует возможность учета расположение уже найденных знаков с предыдущего кадра.

Разработанный алгоритм распознавания и классификации знаков на изображении имеет следующие шаги:

- 1) Предварительная обработка изображения.
- 2) Поиск областей симметрии — мест потенциального расположения знаков.
- 3) Классификация выделенных симметричных областей.

3 Предварительная обработка

Предварительная обработка изображения включает в себя два этапа:

- 1) Перевод исходного изображения в цветовое пространство $L^*a^*b^*$.
- 2) OTSU бинаризация.

Результатом предварительной обработки являются четыре бинаризованных изображения, которые используются для дальнейшего распознавания и классификации знаков на изображении.

3.1 Перевод в цветовое пространство $L^*a^*b^*$

Для анализа изображения оно переводится в цветовое пространство $L^*a^*b^*$, так как оно обладает рядом преимуществ:

- 1) В $L^*a^*b^*$ значение светлоты отделено от значения хроматической составляющей. Светлота хранится в канале L^* . По нему легко определяется в какое время суток получен кадр и в зависимости от этого применять разные методы классификации.
- 2) Хроматические каналы a^* и b^* обозначают положение цвета в разных диапазонах. Так канал a^* обозначает положение цвета в диапазоне от зеленого до красного, а b^* — от синего до желтого.

Таким образом, положительные значения канала a^* соответствуют красному цвету, а отрицательные — зеленому. Для b^* — это синие и желтые цвета. Данную особенность хорошо видно на рисунке 3.1.

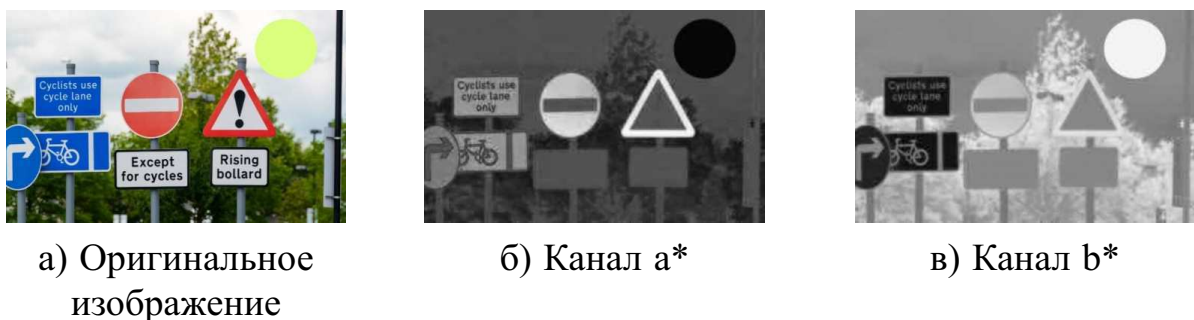


Рисунок 3.1 — Оригинальное изображение после выделения и его хроматические каналы в цветовом пространстве $L^*a^*b^*$

Перевод изображения из цветового пространства RGB в $L^*a^*b^*$ осуществляется в два этапа.

Сначала из RGB изображение переводится в XYZ по формуле (3.1)

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.1)$$

Затем X и Z делятся на 0.950456 и 1.088754 соответственно. После чего вычисляются значения каналов L^* , a^* и b^* по формулам (3.2), (3.4) и (??) соответственно.

$$L = \begin{cases} 116 * Y^{1/3}, & \text{если } Y > 0.008856 \\ 903.3 * Y, & \text{если } Y \leq 0.008856 \end{cases} \quad (3.2)$$

$$a = 500(F(X) - F(Y)) \quad (3.3)$$

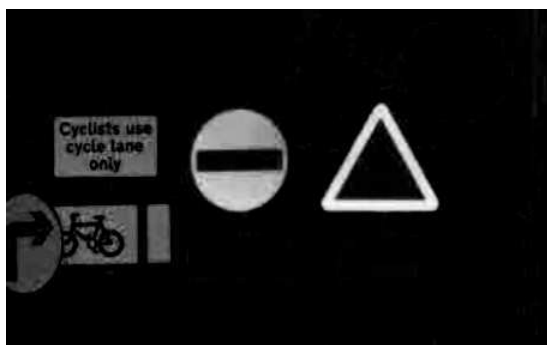
$$b = 200(F(Y) - F(Z)) \quad (3.4)$$

, где

$$f(t) = \begin{cases} t^{1/3}, & \text{если } t > 0.008856 \\ 7.787t + 16/116, & \text{если } t \leq 0.008856 \end{cases}$$

Так как знаки дорожного движения преимущественно красного, синего и желтого цветов, то, для более удобного поиска их на изображении, каждый из хроматических каналов разделяется на отрицательную и положительную части, причем отрицательная часть умножается на -1 . Изображения, полученных в результате данной операции можно увидеть на рисунках 3.2 и 3.3 для каналов a^* и b^* соответственно.

Данное разделение позволяет распределить найденные знаки по цветам для дальнейшей их классификации.



а) Положительная часть



б) Отрицательная часть

Рисунок 3.2 — Положительные и отрицательные составляющие канала a^*



а) Положительная часть



б) Отрицательная часть

Рисунок 3.3 — Положительные и отрицательные составляющие канала b^*

3.2 OTSU бинаризация

Алгоритм OTSU Thresholding[30], после анализа переданного ему изображения, автоматически вычисляет порог бинаризации, который потом используется обычной пороговой бинаризацией. Данный алгоритм позволяет разделить пиксели двух классов («полезные» и «фоновые»), вычисляя такой порог, чтобы внутриклассовая дисперсия была минимальной.

Данный алгоритм использует гистограмму распределения значений яркости изображения для вычисления порога бинаризации[31]. Для изображения строится гистограмма по $p_i = n_i/N$, где i — значение яркости (изменяется от 0 до 255), n_i — количество точек изображения с яркостью i , N — общее количество точек изображения. Диапазон яркости делится на два класса с помощью порогового значения t , которое перебирается от 0 до 255. Для каждого значения t рассчитывается две величины — вероятность принадлежности классу и математическое ожидание

класса. Вероятности вычисляются по формулам (3.5) и (3.6) для первого и второго классов соответственно.

$$w_1(t) = \sum_{i=1}^t p_i \quad (3.5)$$

$$w_2(t) = 1 - w_1(t) \quad (3.6)$$

Математическое ожидание для первого и второго классов вычисляются по формулам (3.7) и (3.7) соответственно.

$$\mu_1(t) = \frac{\sum_{i=1}^t ip_i}{w_1(t)} \quad (3.7)$$

$$\mu_2(t) = \frac{\mu_T - \mu_1(t)w_1(t)}{w_2(t)}, \quad (3.8)$$

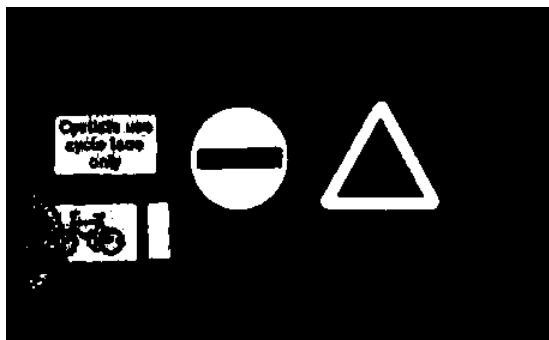
где

$$\mu_T = \sum_{i=0}^{255} ip_i$$

Оценка качества является величина, вычисляемая по формуле (3.9). За итоговый порог бинаризации берется такое t , при котором достигается максимум данной оценки.

$$\sigma^2(t) = w_1(t)w_2(t) [\mu_1(t) - \mu_2(t)]^2, \quad (3.9)$$

В результате бинаризации получаются изображения с ярко выраженным очертанием знаков. На рисунках 3.4 и 3.5 представлены все четыре канала, полученный на прошлом шаге, после бинаризации.



а) Положительная часть

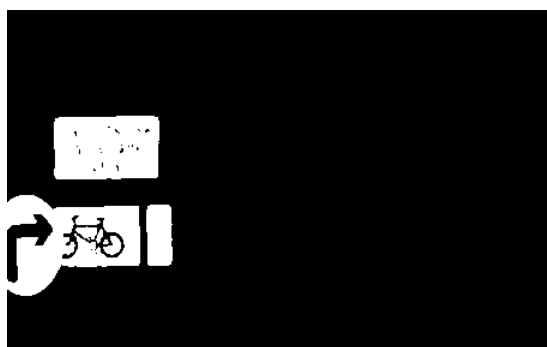


б) Отрицательная часть

Рисунок 3.4 — Бинаризация частей канала a^*



а) Положительная часть



б) Отрицательная часть

Рисунок 3.5 — Бинаризация частей канала b^*

4 Поиск симметричных областей

Так как знаки дорожного движения имеют симметричную форму, то для определения их положения на изображении возможно использовать метод поиска симметрий. Поиск производится во всех четырех изображениях, полученный после предварительной обработки, описанной в главе 3.

В данной работе был выбран быстрый метод поиска радиальной симметрии (fast radial symmetry transform for detecting points of interest), предложенный Гаретом Лойем (Gareth Loy) [32].

4.1 Быстрый метод поиска радиальной симметрии

Преобразование, предложенное Лойем, вычисляется для одного или нескольких радиусов из множества N . Радиусы выбираются в зависимости от того, какого размера форма ищется. Значение преобразования для радиуса $n \in N$ отображает вклад в радиальную симметрию градиента, на расстоянии n от точек изображения. При этом, вычисление для непрерывного множества радиусов обычно не обязательно.

Для каждого радиуса n формируется два изображения:

- Проекция ориентации градиента O_n .
- Проекция магнитуды градиента M_n .

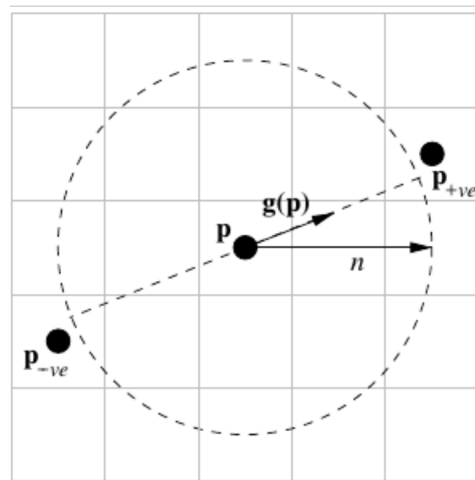


Рисунок 4.1 — Положение точек $p_{+ve}(p)$ и $p_{-ve}(p)$ относительно точки p для $n = 2$

Для получения данных изображений считается градиент g в каждой точке исходного изображения, который определяет точки положительного $p_{+ve}(p)$ и отрицательного $p_{-ve}(p)$ влияния. Точкой положительного влияния является точка, расположенная от заданной точки на расстоянии n по направлению градиента, в то время как точкой отрицательного влияния является точка, расположенная на расстоянии n против направлению градиента. Схематично данные точки показаны на рисунке 4.1

Координаты точек $p_{+ve}(p)$ и $p_{-ve}(p)$ вычисляются по формулам (4.1) и (4.2) соответственно.

$$p_{+ve}(p) = p + \text{round} \left(\frac{g(p)}{\|g(p)\|} n \right) \quad (4.1)$$

$$p_{-ve}(p) = p - \text{round} \left(\frac{g(p)}{\|g(p)\|} n \right) \quad (4.2)$$

, где round означает округление каждого элемента вектора к ближайшему целому числу.

Начальные значения проекции ориентации и магнитуды градиента равны нулю. Затем для каждой точки положительного влияния $p_{+ve}(p)$ увеличиваются значения O_n и M_n в данной точке на 1 и $\|g(p)\|$ соответственно, а для каждой точки отрицательного влияния $p_{-ve}(p)$ — уменьшаются на те же значения:

$$O_n(p_{+ve}(p)) = O_n(p_{+ve}(p)) + 1$$

$$M_n(p_{+ve}(p)) = M_n(p_{+ve}(p)) + \|g(p)\|$$

$$O_n(p_{-ve}(p)) = O_n(p_{-ve}(p)) - 1$$

$$M_n(p_{-ve}(p)) = M_n(p_{-ve}(p)) - \|g(p)\|$$

Искомое преобразование S_n для радиус n определяется по формуле (4.3).

$$S_n = F_n * A_n \quad (4.3)$$

, где

$$F_n(p) = ||\tilde{O}_n(p)||^{(\alpha)} \tilde{M}_n(p), \quad (4.4)$$

$$\tilde{O}_n(p) = \frac{O_n(p)}{\max_p \{||O_n(p)||\}},$$

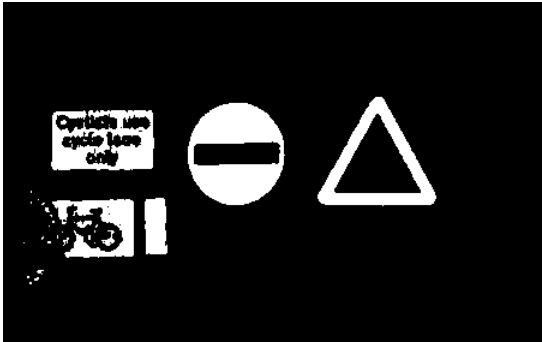
$$\tilde{M}_n(p) = \frac{M_n(p)}{\max_p \{||M_n(p)||\}},$$

α — параметр радиальной строгости, A_n — ядро Гауссиана, $*$ — операция свертки.

Полное преобразование S определяется как сумма преобразований для выбранных радиусов N :

$$S = \sum_{n \in N} S_n$$

Таким образом положительные значения в S соответствуют ярким областям радиальной симметрии, а отрицательные — темным. Пример преобразования можно увидеть на рисунке 4.2.



а) Положительная часть канала
а* после бинаризации



б) Изображение после
преобразования

Рисунок 4.2 — Пример работы быстрого метода поиска радиальной симметрии

Параметр α в формуле (4.4) определяет насколько строго радиальными будут найденные области. Для большинства приложений достаточно $\alpha = 2$. Чем выше данный параметр, тем более строго радиальными будут найденные области.

Значение дисперсии σ для ядра Гауссианы в данной работе равно $0.25n$.

4.2 Оптимизация

Для вычисления полного преобразования нет необходимости в последовательном вычислении преобразований для каждого радиуса n . Таким образом, данный метод можно ускорить, если запускать вычисления для радиусов одновременно в разных потоках, а потом, после их завершения, вычислять итоговый результат.

В данной работе для поиска симметричных форм использовались радиусы размерами 10, 15, 25. Были проведены замеры времени работы метода с и без включения данной оптимизации. Результаты (таблица 4.1) показали ускорение работы алгоритма в полтора-два раза.

Таблица 4.1 — Время работы метода в зависимости от размера изображения и наличия оптимизации

Размер изображения	Время работы без оптимизации, мс	Время работы с оптимизацией, мс
108x193	15	10
404x648	156	95
420x420	95	77
480x704	195	136
512x512	175	120
600x800	250	153
1080x1920	1277	595
2304x3072	4820	2195
2592x3872	6704	3219

5 Классификация знаков дорожного движения

В качестве классификатора был использован метод опорных векторов (SVM). Обучение классификатора проходило на открытой базе российских знаков. Для каждого знака дорожного движения был обучен свой классификатор.

5.1 Использование цвета

Так как классификация является трудоемким процессом, а самих классификаторов — много (по одному на каждый класс), то для ускорения процесса классификации было решено разделить все классификаторы по группам, в зависимости от цвета знака, для которого они обучены. Это позволило проверять найденные потенциальные знаки не на всех классификаторах, а только на тех, которые соответствуют цвету знака.

В качестве цветов групп были выделены: красный, синий, зеленый и желтый.

5.2 Deskриптор интересных точек

Для работы классификатора SVM происходит не с самим изображением, а с дескрипторами интересных точек. От выбора типа дескрипторов зависит скорость и качество классификации. В ходе работы было рассмотрены SIFT и HOG дескрипторы.

5.2.1 SIFT-дескрипторы

SIFT — один из методов определения и описания точек интереса изображений, а так де нахождения их дескрипторов. Данный дескриптор инвариантен поотношению к следующим преобразованиям:

- 1) Смещение.
- 2) Поворот.
- 3) Масштаб.
- 4) Изменения яркости.
- 5) Изменения положения камеры.

Основным в детектировании особых точек является построение пирамиды гауссианов (Gaussian) и разностей гауссианов (Difference of Gaussian, DoG). Гауссианом (или изображением, размытым гауссовым фильтром) является изображение: $L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$, где L — значение гауссиана в точке с координатами (x, y) , σ — радиус размытия, G — гауссово ядро, I — значение исходного изображения, $*$ — операция свертки.

Разностью гауссианов называют изображение, полученное путем попиксельного вычитания одного гауссиана исходного изображения из гауссиана с другим радиусом размытия. $D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma)$, где D — значение разности гауссианов в точке.

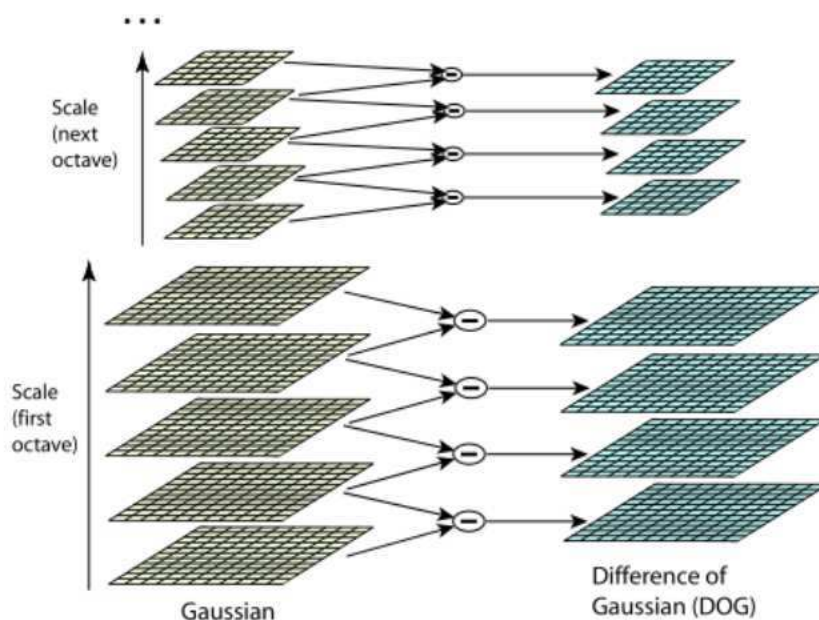


Рисунок 5.1 — Пирамида гауссианов (слева) и их разность (справа)

Инвариантность относительно масштаба достигается за счет нахождения ключевых точек для исходного изображения, взятого в разных масштабах. Для этого строится пирамида гауссианов: все масштабируемое пространство разбивается на некоторые участки — октавы, причем часть масштабируемого пространства, занимаемого следующей октавой, в два раза больше части, занимаемой предыдущей. К тому же, при переходе от одной октавы к другой размеры изображения уменьшаются вдвое.

Параллельно с построением пирамиды гауссианов, строится пирамида разностей гауссианов, состоящая из разностей соседних изображений в пирамиде гауссианов.

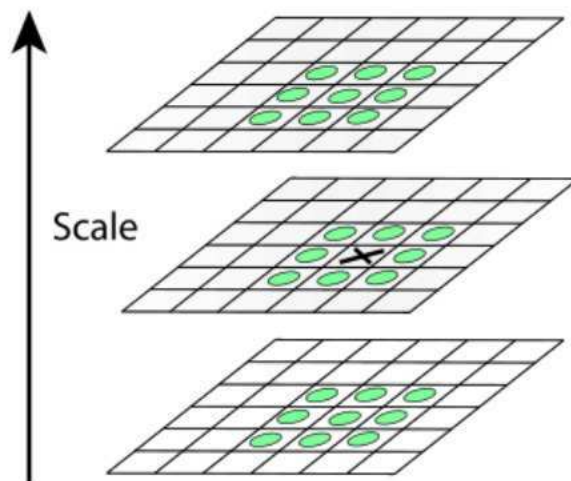


Рисунок 5.2 — Пирамида гауссианов (слева) и их разность (справа)

На рисунке 5.1 схематично показано, что каждая разность получается из двух соседних гауссианов, количество разностей на единицу меньше количества гауссианов, при переходе к следующей октаве размер изображений уменьшается вдвое.

После построения пирамид ищутся точки интереса. Будем считать точку интересной, если она является локальным экстремумом разности гауссианов (рисунок 5.2).

В каждом изображении из пирамиды DoG ищутся точки локального экстремума. Каждая точка текущего изображения DoG сравнивается с её восьмью соседями и с девятью соседями в DoG, находящимися на уровень выше и ниже в пирамиде. Если эта точка больше (меньше) всех соседей, то она принимается за точку локального экстремума.

Для каждой из полученной интересной точки происходит ее уточнение и нахождение ориентации.

Направление ключевой точки вычисляется исходя из направлений градиентов точек, соседних с особой. Все вычисления градиентов производятся на изображении в пирамиде гауссианов, с масштабом наиболее близким к масштабу ключевой

точки. Величина и направление градиента в точке (x, y) вычисляются по формулам (5.1) и (5.2) соответственно.

$$m(x, y) = \sqrt{[L(x + 1, y) - L(x - 1, y)]^2 + [L(x, y + 1) - L(x, y - 1)]^2} \quad (5.1)$$

$$\theta(x, y) = \tan^{-1} \left(\frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)} \right) \quad (5.2)$$

Направление ключевой точки находится из гистограммы направлений O .

Дескриптор — идентификатор ключевой точки, выделяющий её из остальной массы особых точек. В методе SIFT дескриптором является вектор. Этот вектор вычисляется на гауссиане, ближайшем по масштабу к ключевой точке, и исходя из градиентов в некотором окне ключевой точки. Перед вычислением дескриптора это окно поворачивают на угол направления ключевой точки, чем и достигается инвариантность относительно поворота.

После получения дескрипторов они нормализуются, после чего все его компоненты, значение которых больше 0.2, урезаются до значения 0.2 и затем дескриптор нормализуется ещё раз. После этого дескриптор готов к использованию.

5.2.2 Гистограмма направленных градиентов (HOG)

Гистограмма направленных градиентов (Histogram of Oriented Gradients, HOG) — это дескрипторы особых точек, которые используют для распознавания объектов. Дескриптор HOG, так же как и SIFT, является инвариантным к геометрическим и фотометрическим преобразованиям.

Вычисление дескриптора происходит следующим образом:

- 1) Вычисление градиента. Распространенным методом является применение одномерной маски в горизонтальном и вертикальном направлении.
- 2) Разделение изображения на маленькие связные области (ячейки).
- 3) Вычисляются гистограммы ячеек. Каждый пиксел в ячейке участвует во взвешенном голосовании для каналов гистограммы направлений, основан-

ном на значении градиентов. Каналы гистограммы равномерно распределяются от 0 до 180. Вес пикселя задает абсолютное значение градиента.

4) Градиенты локально нормируются, после чего ячейки группируются в более крупные связные блоки. Deskриптор HOG, таким образом, является вектором компонент нормированных гистограмм ячеек из всех областей блока.

5) Нормализация блока.

Размер полученного deskриптора зависит от выбранных размеров ячейки и блока.

Преимуществом данного deskриптора по сравнению с deskрипторами SIFT является то, что результатом является один вектор, описывающий интересные точки изображения. Метод SIFT подразумевает deskриптор для каждой интересной точки. Это увеличивает время работы классификатора, поэтому в данной работе были выбраны HOG-desкрипторы.

5.3 Метод опорных векторов (SVM)

Метод опорных векторов (или SVM, support vector machine) — это набор алгоритмов для обучения с учителем, использующихся для задач классификации [33].

Основная идея метода — перевод исходных исходных данных в пространство более высокой размерности, где ищется такая гиперплоскость, которая разделяет исходные классы и минимизирующая среднюю ошибку классификатора.

Более формально — для каждого объекта задается набор пар: (x_i, c_i) , где x_i — вектор, характеризующий данный объект, а c_i — число, определяющее класс данного объекта.

Затем для всех наборов (x_i, c_i) выборки обучается классификатор, который может быть в дальнейшем использован.

В данной работе каждый классификатор, относящейся к своему знаку дорожного движения, был обучен на двух классах: «знак» и «не знак». Для обучения в качестве примеров класса «знак» были использованы доступные изображения дан-

ного знака, а в качестве примеров класса «не знак» были использованы как другие знаки, так и случайные части кадров видео с видеорегистратора.

6 Состав и структура системы

Видео поток, полученный с автомобильного видеорегистратора является непрерывным, а значит что при обработке очередного кадра видео возможно использование информации, полученной ранее.

Основная структура системы показана на рисунке 6.1.

После предварительной обработки, описанной в главе 3, изображение передается в модули распознавания и слежения. Оба модуля используют быстрый метод поиска радиальной симметрии для определения положения и размера «потенциальных» знаков. Модуль распознавания ищет новые симметричные формы («потенциальных» знаки) небольшого радиуса в видеопотоке. Модуль слежения — для слежением за ранее найденными формами.

По найденным центрам и радиусам вырезается часть исходного изображения для дальнейшей обработки классификатором.

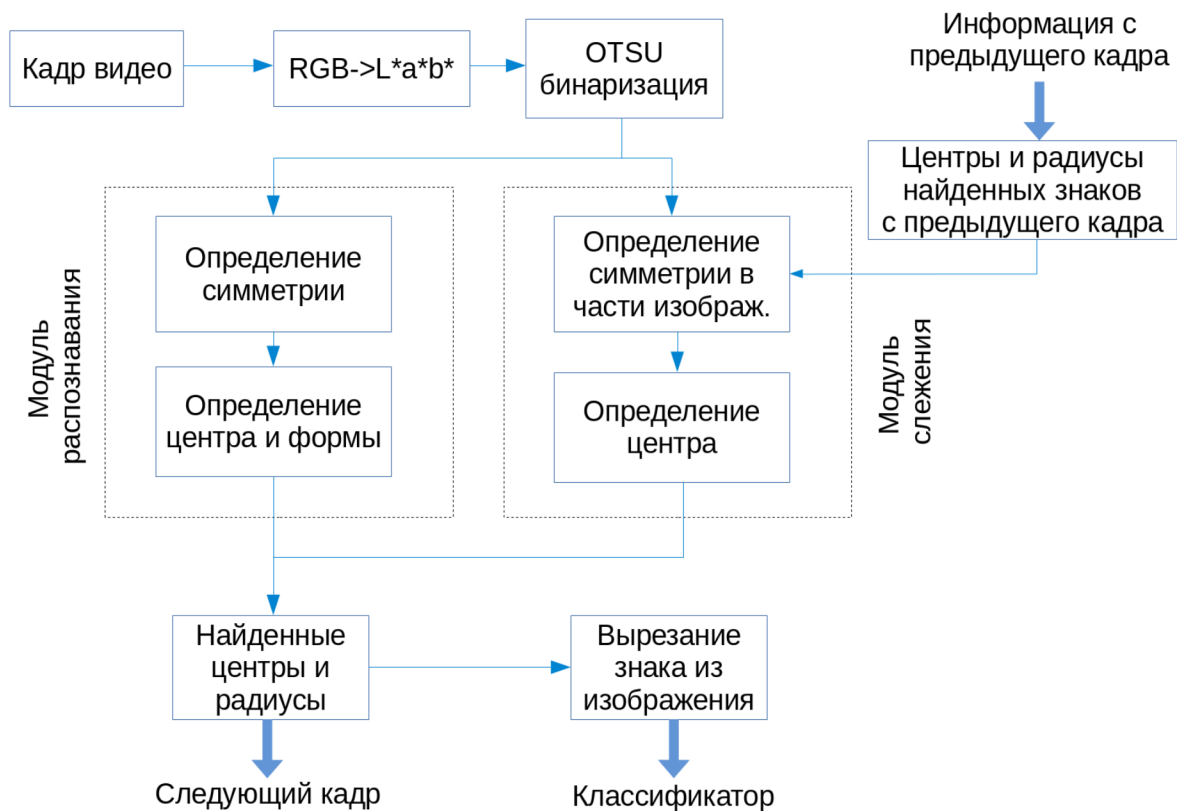


Рисунок 6.1 — Основная структура системы

6.1 Модуль распознавания

Данный модуль используется для нахождения радиально симметричных форм небольшого размера в кадре. Для этого используется алгоритм, описанный в главе 4.

В качестве радиусов были выбраны значения 10, 15 и 20 пикселей. Поиск осуществляется во всех четырех хроматических каналах. Для каждой найденной формы запоминается ее центр, радиус и класс, если знак смог классифицироваться. Данная информация будет передана в модуль слежения во время обработки следующего класса.

6.2 Модуль слежения

По мере приближения автомобиля к установленному знаку, он увеличивается, а это значит, что для успешного определения данного знака модулем распознавания необходимо увеличивать количество радиусов для поиска. В свою очередь, это приведет к значительному увеличению времени классификации знаков на видео кадрах.

Для решения данной проблемы был разработан модуль слежения. Задачей данного модуля является отслеживание передвижения уже найденных знаков.

Для каждого из найденных знаков модуль хранит его радиус симметрии и прямоугольная область в которой расположен знак.

При обработке очередного кадра перебираются все знаки, хранимые модулем, и для каждого из них выполняются следующие действия:

- 1) Прямоугольная область данного знака увеличивается на 10 пикселей в каждом направлении.
- 2) В полученной новой области производится поиск симметрии с радиусом на единицу больше, чем радиус симметрии текущего знака.
- 3) Найденный знак классифицируется.

Если после классификации класс остался там же, то область знака и радиус обновляются, а знак будет использован при обработке следующих кадров.

Если после классификации определился другой знак или ни один не определился, то текущий знак помечается как вероятно скрывшимся и после повторения ситуации в следующих пяти подряд идущих кадрах данный знак удаляется из дальнейшего рассмотрения.

На рисунках 6.2, 6.3 и 6.4 изображен знак, определенный модулем слежения.

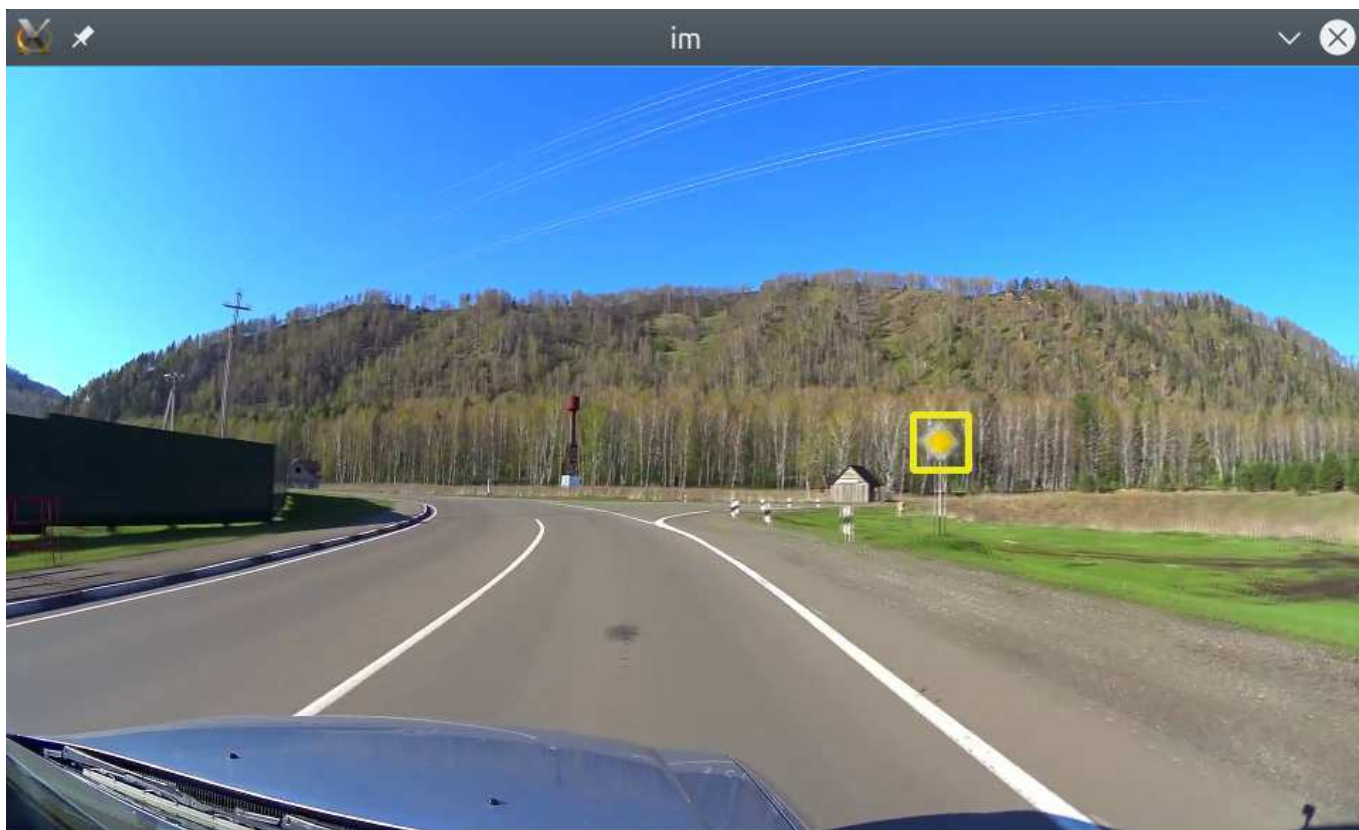


Рисунок 6.2 — Распознанный и классифицированный знак

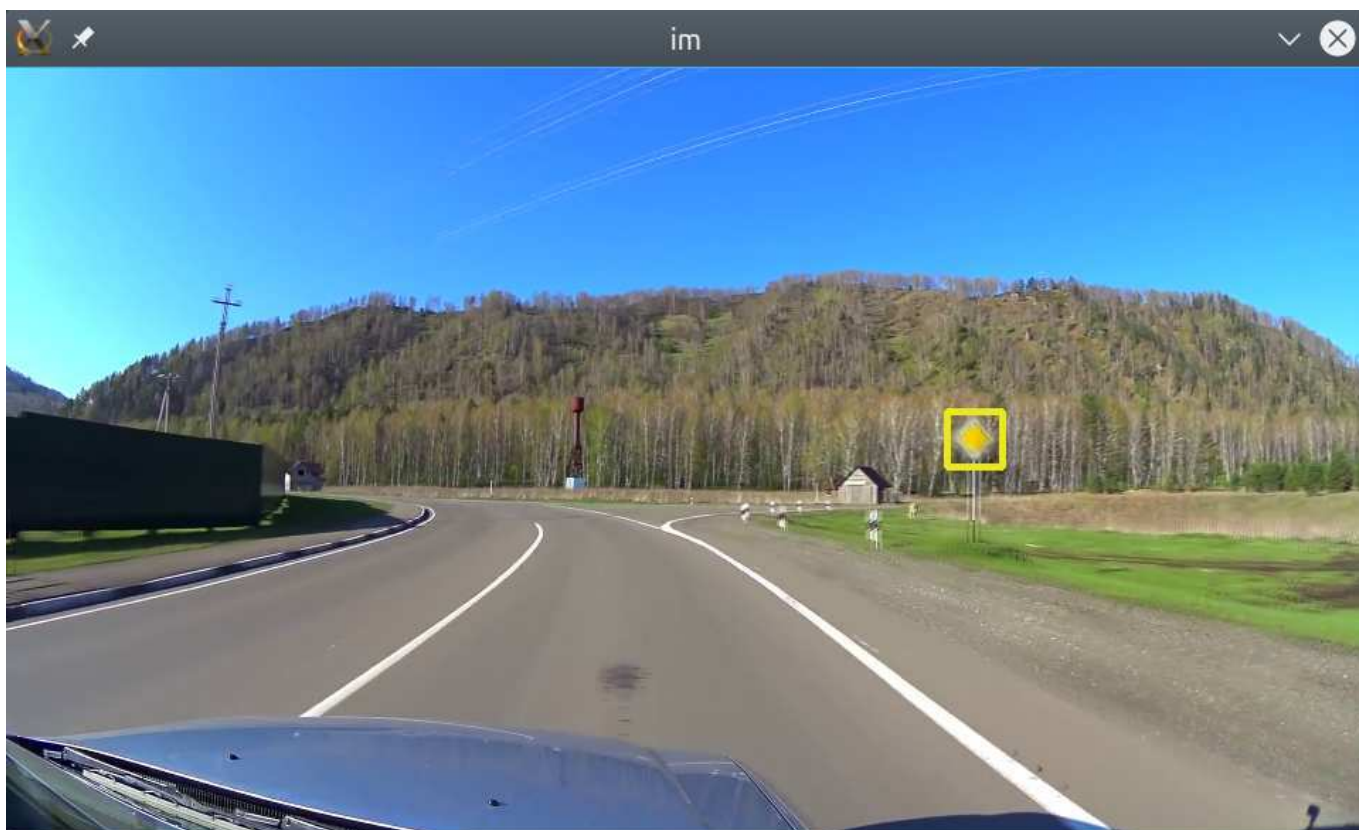


Рисунок 6.3 — Распознанный и классифицированный знак

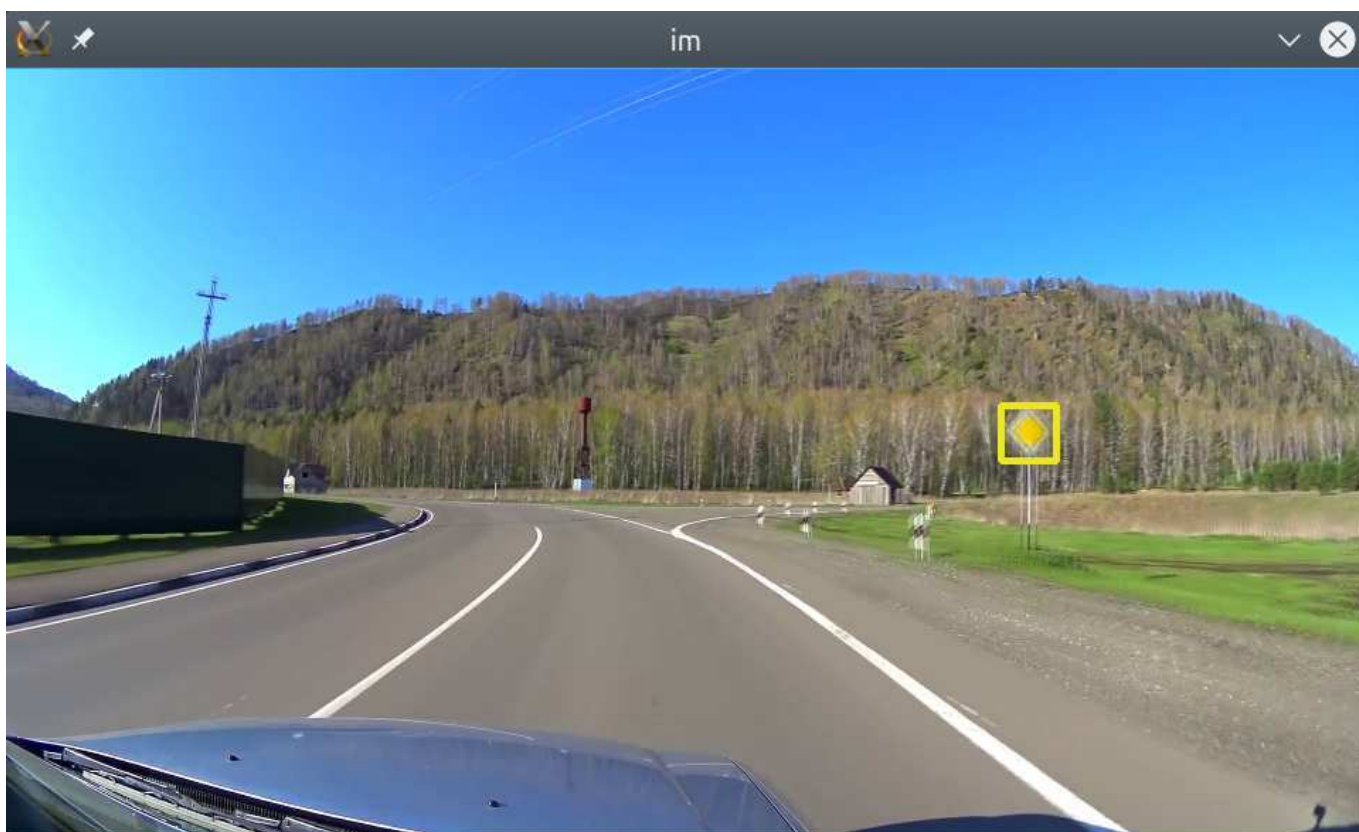


Рисунок 6.4 — Распознанный и классифицированный знак

7 Описание программного обеспечения

Для разработки системы был выбран язык C++. Так же была использована библиотека OpenCV для работы с изображениями.

7.1 Описание основных разработанных классов

Класс: `Sample`

Описание: Класс, хранящий в себе информацию о найденной симметричной форме (или знаке). Так же используется для обучения классификатора.

Поля:

- 1) `cv::Mat mat` — хранимая часть исходного изображения.
- 2) `cv::Rect roi` — позиция и размер `mat` в исходном изображении.
- 3) `int type` — класс распознанного знака. По умолчанию равен `Sample::UNKNOWN_TYPE`.

Методы:

- 1) `Sample(cv::Mat m, cv::Rect roi, int type = UNKNOWN_TYPE)` — конструктор с параметрами, инициализирующими класс. Параметры совпадают с полями класса.

Класс: `CDataSetParser`

Описание: Класс для парсинга базы знаков учебной выборки.

Поля: Отсутствуют

Методы:

- 1) `static SamplesT CDataSetParser::ParsePPMS(const std::string& _dir_path, const std::string& _file_name)` — метод, возвращаемый вектор найденный обучающих примеров.

Параметры:

- `const std::string& _dir_path` — путь до папки с базой.
- `const std::string& _file_name` — имя файла списка выборки.

Возвращаемое значение: вектор найденный обучающих примеров.

- 2) **static** SamplesT CDataSetParser::ParseLIS(**const** std::string& _dir_path) — метод, возвращаемый вектор найденный тестовых примеров.

Параметры:

- **const** std::string& _dir_path — путь до папки с базой.

Возвращаемое значение: вектор найденный тестовых примеров.

Класс: CFRSD

Описание: Класс реализации алгоритма Fast Radial Symmetry Transform for Detecting Points of Interest).

Поля:

- 1) std::vector<**int**> m_Radii; — список радиусов, для которых вычисляется преобразование.
- 2) **float** m_Alpha; — параметр радиальной строгости.

Методы:

- 1) **static** FRSDPtrT create(**const** std::vector<**int**>& _radii, **float** _alpha = 2); — фабричный метод. Создает экземпляр с переданными параметрами.

Параметры:

- **const** std::vector<**int**>& _radii — список радиусов, для которых вычисляется преобразование.
- **float** _alpha = 2 — параметр радиальной строгости, значение по умолчанию: 2.

Возвращаемое значение: экземпляр класса CFRSD с установленными параметрами.

- 2) CFRSD(**const** std::vector<**int**>& _radii, **float** _alpha); — конструктор с параметрами, инициализирующими класс.

Параметры:

- **const** std::vector<**int**>& _radii — список радиусов, для которых вычисляется преобразование.

– **float** _alpha — параметр радиальной строгости.

3) `cv::Mat calculate(cv::Mat _src);` — вычисляет преобразование.

Параметры:

– `cv::Mat _src` — переданно изображение для которого будет посчитано преобразование.

Возвращаемое значение: Преобразование.

Функции:

1) `void calculate_impl(cv::Mat _src, int r, float alpha, cv::Mat& _dst);` — функция, реализующая алгоритм для одного радиуса.

Класс: `CTracker`

Описание: Класс, предназначенный для слежения за найденными знаками.

Поля:

1) `SampleT m_Samples;` — семплы с предыдущего шага.

2) `FRSDPtrT m_Frsd;`

Методы:

1) `void NextFrame(cv::Mat _image, const SampleT& _samples);` — через этот метод осуществляется информирование трекера о новом кадре.

Параметры:

– `cv::Mat _image` — кадр изображения.

– `const SampleT& _samples` — найденные небольшие области симметрии для данного кадра.

2) `cv::Rect CalculateWindow(const Sample& _sample);` — вычисляет окно поиска для данного семпла.

Параметры:

– `const Sample& _sample` — семпл.

3) `cv::Mat ApplyFRSD(const Sample& _sample, cv::Rect _roi);` — применяет FRSD в прямоугольнике.

Параметры:

– `cv::Mat _image` — кадр изображения.

- **const** SamplesT& _samples — найденные небольшие области симметрии для данного кадра.

4) SamplesT GetSamples(); — получение списка семплов.

Класс: CClassifier

Описание: Класс предназначенный для выполнения классификации знаков.

Поля:

- 1) SamplesT m_Samples; — список семплов.
- 2) std::map<int, cv::Ptr<cv::ml::SVM>> m_SVM; — классификаторы для каждого типа знака.
- 3) std::vector<cv::Mat> m_Descriptors; — дескрипторы семплов.

Методы:

- 1) CClassifier(); — конструктор.
- 2) **void** Load(**const** std::string& _file_path); — метод для загрузки данных классификаторов из файла.

Параметры:

- **const** std::string& _file_path — путь до файла.

- 3) **void** Save(**const** std::string& _file_path); — метод для сохранения данных классификаторов в файл.

Параметры:

- **const** std::string& _file_path — путь до файла.

- 4) **void** AddSample(**const** Sample& _sample); — метод, предназначенный для добавления одного семпла для дальнейшей обработки.

Параметры:

- **const** Sample& _sample — семпл для добавления.

- 5) **void** AddSamples(**const** SamplesT& _samples); — метод, предназначенный для добавления списка семплов для дальнейшей обработки.

Параметры:

- **const** SamplesT& _samples — список семплов для добавления.

6) **void** Train(); — метод, обучающий классификаторы на семплах, добавленные ранее.

7) **SampleT** Predict(); — метод, классифицирующий добавленные ранее семплы.

Возвращаемое значение: семплы с выслевленным классом.

8) **const** **SampleT&** GetSamples() — метод для получения текущего списка семплов.

Возвращаемое значение: список семплов в классификаторе.

9) **void** AddDescriptor(**const** **Sample&** _sample); — метод, создающий дескриптор для семпла.

Параметры:

— **const** **Sample&** _sample — семпл для которого необходимо вычислить дескриптор.

10) **int** Predict(**cv::Mat** m); — метод, классифицирующий изображение и возвращаемы найденный класс.

Параметры:

— **cv::Mat** m — изображения, которое необходимо классифицировать.

Возвращаемое значение: найденный класс.

Класс: **CDetector**

Описание: Класс, предназначенный для предварительной обработки изображения и запуска быстрого алгоритма поиска радиальной симметрии.

Поля:

1) **cv::Mat** m_SourceImage; — исходное изображение.

2) **std::vector<cv::Mat>** m_LabChannels; — каналы L^* , a^* и b^* исходного изображения в цветовом пространстве Lab.

3) **std::vector<cv::Mat>** m_NPChannels; — положительные и отрицательные части каналов a^* и b^* .

- 4) `std::vector<cv::Mat> m_FRSDChannels;` — положительные и отрицательные части каналов a^* и b^* после алгоритма быстрого поиска радиально симметричных форм.

Методы:

- 1) **static** `DetectorPtrT create(cv::Mat _image, const std::vector<int>& _radii, float _alpha = 2);` — статический метод для создания детектора.

Параметры:

- `cv::Mat _image` — изображение, текущий кадр.
- **const** `std::vector<int>& _radii` — радиусы которые используются для поиска симметрии.
- **float** `_alpha = 2` — значение радиальной строгости алгоритма.

- 2) **static** `DetectorPtrT create(cv::Mat _image, FRSDPtrT _frsd);`

Параметры:

- `cv::Mat _image` — изображение, текущий кадр.
- `FRSDPtrT _frsd` — объект для поиска симметрии.

- 3) **const** `std::vector<cv::Mat> GetLabChannels()const;` — метод для получение каналов $L^*a^*b^*$.

Возвращаемое значение: список каналов.

- 4) **const** `std::vector<cv::Mat> GetNPChannels()const;` — метод для получение отрицательных и положительных частей каналов a^* и b^* .

Возвращаемое значение: список каналов отрицательных и положительных частей каналов a^* и b^* .

- 5) **const** `std::vector<cv::Mat> GetFRSDChannels()const;` — метод для получение отрицательных и положительных частей каналов a^* и b^* после алгоритма FRSD.

Возвращаемое значение: список каналов отрицательных и положительных частей каналов a^* и b^* после алгоритма FRSD.

- 6) `CDetector(cv::Mat _image);` — конструктор класса.

Параметры:

– `cv::Mat _image` — исходное изображение.

7) **void** `SplitChannels()`; — метод для разделения изображения на каналы `L*a*b*`.

8) **void** `ApplyFRSD(FRSDPtrT _frsd)`; — применение FRSD.

Параметры:

– `FRSDPtrT _frsd` — объект для поиска симметрии.

Заключение

В процессе выполнения данной работы были исследованы основные проблемы, возникающие в задачах компьютерного зрения. Определены основные цели и задачи, а так же выполнен обзор существующих подходов и алгоритмов, использующихся для распознавания и классификации различных образов и знаков дорожного движения в частности. Так же был реализован быстрый метод поиска радиальной симметрии.

В результате выполнения поставленных был спроектирован и реализован программный комплекс, позволяющий распознать и классифицировать дорожные знаки в видео потоке.

Планируется дальнейшая работа над поставленными задачами, оптимизация реализованных алгоритмов и увеличение точности классификации дорожных знаков.

Данная работа была представлена на XVI Всероссийской конференции молодых ученых по математическому моделированию и информационным технологиям [34].

Список использованных источников

1. Estevez L. A real-time histographic approach to road sign recognition / L. Estevez, N. Kehtarnavaz // IEEE Southwest Symposium on Image Analysis and Interpretation. — 1996.
2. Loy G. Fast shape-based sign detection for a drives assisatnce system / G. Loy, N. Barnes // IEEE/RSJ International Conference on Intelligent Robots and Systems. — 2004.
3. Aoyagi Y. A study on traffic sign recognition in scene image using genetic algorithms and neural networks / Y. Aoyagi, T. Asakura // The1996 IEEE IECON 22nd Inter. Conf. on Industrial Electronics, Control and Instrumentation. — 1996.
4. Escalera A. Traffic sign recognition and analysis for intelligent vehicles / A. Escalera, J. Armingol, M. Mata // Image and Vision Comput. Vol. 21. — 2003. — Pp. 247
5. Aoyagi Y. Multi-feature Hierarchical template matching using distance transforms / Y. Aoyagi, T. Asakura // The 35th SICE Annual Conference. — 1996.
6. Gavrilă D. Multi-feature Hierarchical template matching using distance transforms / D. Gavrilă // Fourteenth Inter. Conf. on Pattern Recognition. — 1998.
7. Schiek C. A fast traffic sign recognition algorithm for gray value images / C. Schiek // 8th Inter. Conf. Computer Analysis of Images and Patterns. — 1999.
8. Богущ Р. П. Обнаружение и локализация объектов на статических изображениях с использованием минимаксных функций схожести / Р. П. Богущ // Вестник Полоцкого государственного университета. — 2007. — С. 10 —114.
9. Puntavongkour S. Automatic Recognition and location of road signs from terrestrial color imagery / S. Puntavongkour, X. Chen, M. Kusanagi // Geoinformatics and DMGIS. — 2001.

10. Hirose K. Real-time recognition of road traffic sign in moving scene image using new image filter / K. Hirose, T. Asakura, Y. Aoyagi // 26th Annual Conf. IEEE Industrial Electronics Society. — 2000.
11. Liu H. Real-time recognition of road sign in motion image based on genetic algorithm / H. Liu, D. Liu, J. Xin // 2002 Inter. Conf. Machine Learning and Cybernetics. — 2002.
12. Liu H. Real-time recognition of road traffic sign in moving scene image using genetic algorithm / H. Liu, D. Liu, Q. Li // 4th World Congress on Intelligent Control and Automation. — 2002.
13. Loy G. Fast shape-based sign detection for a drives assisatnce system / G. Loy, N. Barnes // IEEE/RSJ International Conference on Intelligent Robots and Systems. — 2004.
14. Piccioli G. Robust method for sign detection and recognition. Vol. 14 / G. Piccioli — 1996. — Pp. 209–223.
15. Фурсов В. А. Локализация контуров объектов на изображениях при вариациях масштаба с использованием преобразования Хафа / В. А. Фурсов, С. А. Бибилов, П. Ю. Якимов // Компьютер. оптика. Т. 37. — 2013. — С. 49 —502.
16. Vitabile S. Pictogram road signs detection and understanding in out door scenes / S. Vitabile, F. Sorbello // Conf. Enhanced and Synthetic Vision. — 1998.
17. Ohara H. Detection and recognition of road signs using simple layered neural network / H. Ohara // The 9th Inter. Conf. Neural Information Processing. — 2002.
18. Puntavungkour S. Automatic Recognition and location of road signs from terrestrial color imagery / S. Puntavungkour, X. Chen, M. Kusanagi // Geoinformatics and DMGIS. — 2001.
19. Nakamura M. Searching and recognition of road signpost using ring detetion network / M. Nakamura // 1999 IEEE Inter. Conf. on Systems, man, and Cybemetics. — 1999.

20. Kellmeyer D. Detection of highway warning signs in natural video images using color image processing and neural networks / D. Kellmeyer, H. Zwahlen // 1994 IEEE World Congress on Computational Intelligence. — 1994.
21. Ghica D. Recognition of traffic signs by artificial neural network / D. Ghica, S. Lu, X. Yuan // IEEE Inter. Conf. Neural Networks. — 1994.
22. Aoyagi Y. A study on traffic sign recognition in scene image using genetic algorithms and neural networks / Y. Aoyagi, T. Asakura // The 1996 IEEE IECON 22nd Inter. Conf. on Industrial Electronics, Control and Instrumentation. — 1996.
23. Харинов М. В. Модель локализации объектов на цифровом изображении [Текст] / М. В. Харинов // Вестник бурятского государственного университета. - 2013. - С. 182— 189..
24. Escalera A. Traffic sign recognition and analysis for intelligent vehicles / A. Escalera, J. Armingol, M. Mata // Image and Vision Comput. Vol. 21. — 2003. — Pp. 247
25. Escalera A. Visual sign information extraction and identification by deformable models for intelligent vehicles / A. Escalera, J. Armingol, J. Pastor // IEEE Trans. on Intelligent Transportation Systems. Vol. 5. — 2004. — Pp. 57–68.
26. Nakamura M. Searching and recognition of road signpost using ring detection network / M. Nakamura // 1999 IEEE Inter. Conf. on Systems, man, and Cybernetics. — 1999.
27. Lafuente-Arroyo S. Traffic sign shape classification evaluation I: SVM using distance to borders / S. Lafuente-Arroyo // IEEE Intelligent Vehicles Symposium. — 2005.
28. Gielis curves - Road Sign Recognition System [Электронный ресурс]. — Режим доступа: <https://sites.google.com/site/mcvibot2011sep/background/gielis-curves>, свободный
29. Superformula [Электронный ресурс]. — Режим доступа: <https://en.wikipedia.org/wiki/Superformula>, свободный

30. Otsu N. A threshold selection method from gray-level histograms / N. A Otsu // IEEE Transactions on System Man Cybernetics. — 1979. — Pp. 62–66.
31. Bradski G. Learning OpenCV : Computer Vision with the OpenCV Library. — O'Reilly MediaLondon, 2008. — P. 580.
32. Loy G. Fast Radial Symmetry for Detecting Points of Interest / G. Loy, A. Zelinsky // IEEE Transactions on Pattern Analysis and Machine Intelligence. — August 2003. — Vol. 25. — Pp. 959–973.
33. Cristianini N. An Introduction to Support Vector Machines and other kernel-based learning methods. — 2000.
34. Уваров Д. А. Система автоматического распознавания дорожных знаков // Тезисы докладов XVI Всероссийской конференции молодых ученых по математическому моделированию и информационным технологиям. — Красноярск : Институт вычислительных технологий СО РАН, 2015. — С. 9 —95. — Режим доступа: <http://conf.nsc.ru/files/conferences/ym2015/298631/YM2015.pdf>.

Приложение А

Задание на магистерскую диссертацию

Министерство образования и науки Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Алтайский государственный технический университет им. И.И. Ползунова»

УТВЕРЖДАЮ

Заведующий кафедрой С. А. Кантор
«31» 12 2016 г.
подпись и. о. фамилия

ЗАДАНИЕ № 4

НА ДИПЛОМНОЕ ПРОЕКТИРОВАНИЕ

По специальности Программная инженерия

студенту группы 8ПИ-41

Уваров Дмитрий Андреевич

фамилия, имя, отчество

Тема: Разработка системы автоматического распознавания дорожной обстановки в видеопотоке

Утверждено приказом ректора от 31.12.2015 № Л – 4602

Срок исполнения проекта до 12.06.2016

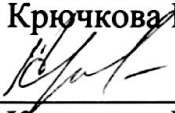
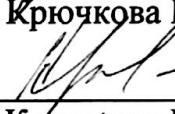

Задание принял к исполнению Уваров Д. А.
подпись фамилия, имя, отчество

Барнаул 2016

1 Исходные данные

Задание на выполнение, техническая документация для OpenCV, ресурсы Internet, ГОСТы, руководящие документы

2 Содержание разделов проекта

Наименование и содержание разделов проекта	Трудоемкость, % от всего объема проекта	Срок выполнения	Консультант (Ф.И.О., подпись)
1 Предметная область	30	20.03.2016	Крючкова Е.Н. 
2 Проектирование	35	19.04.2016	Крючкова Е.Н. 
3 Реализация	35	25.05.2016	Крючкова Е.Н. 

3 Научно-библиографический поиск

3.1 По научно-технической литературе просмотреть РЖ «Кибернетика», «Программное обеспечение» за последние 2 года и научно-технические журналы «Программная инженерия», «Программирование» за последние 2 года.

3.2 По нормативной литературе просмотреть указатели государственных и отраслевых стандартов за последний год.

3.3 Патентный поиск провести за 4 года по странам Россия

Руководитель проекта


подпись

Е.Н. Крючкова
и., о., фамилия